Dublin City University

Faculty of Engineering and Computing

School of Electronic Engineering

This dissertation is submitted for the degree of Doctor of Philosophy

A Machine Learning Approach to the Unsupervised Segmentation of Mitochondria in Subcellular Electron Microscopy Data

by

Julia DIETLMEIER, MSc

Supervisor: Prof. Paul F. WHELAN

2017

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed:.....Julia Dietlmeier..... (Candidate)

ID No.: 58105727

Date:.....13 December 2016.....

Acknowledgements

First of all, I would like to thank my academic advisor Professor Paul F. Whelan for giving me the unique opportunity to work on this challenging interdisciplinary research project, for his professional support during my PhD study, for his patience, motivation and immense knowledge. His guidance helped me in all the time of research, manuscript preparation and revision stages in the submission process to the selected peer-reviewed journals and conferences and the planning and writing of this thesis.

I would also like to express my gratitude to Dr. Ovidiu Ghita for sharing his expertise and research insights which helped me tremendously in directing the course of my work. While working together before the deadlines, his comments, encouragement, project management skills as well as critical appraisement helped me to widen my research from various perspectives.

I wish to thank many current and former colleagues from DCU and from the Center for Image Processing and Analysis (CIPA). It has been a pleasure discussing and sharing different experiences with them.

I sincerely thank the external examiner Prof. Xianghua Xie (Swansea University, UK) and the internal examiner Dr. Noel Murphy (DCU, Ireland) for critically reading the thesis and suggesting substantial improvements.

I would like to extend my thanks to the National Biophotonics and Imaging Platform Ireland (PRTLI Cycle 4) for supporting financially my research program.

Last but not the least, I am grateful to my husband, my children Brian and Emily, my parents and all the extended family for their permanent encouragement, inspiration and support during this research journey. This accomplishment would not have been possible without them.

Contents

1 Introduction **1**
1.1 Subcellular Imaging and Electron Microscopy . . . 1
1.2 Challenges in Mitochondria Segmentation and Apoptosis . . . 2
1.3 Advances in Cellular Image Segmentation . . . 7
1.4 Overview of Objectives and Contributions . . . 8
1.5 Publications Emerging from this Research . . . 10

2 Background Research **12**
2.1 Subcellular Segmentation . . . 13
2.2 Machine Learning for Biomedical Imaging . . . 15
2.2.1 Supervised Machine Learning . . . 16
2.2.2 Neural Networks . . . 17
2.2.2.1 Perceptron . . . 17
2.2.2.2 Multi Layer Perceptron . . . 19
2.2.2.3 Convolutional Neural Networks and Deep Learning . . . 21
2.2.2.4 Neocognitron . . . 22
2.2.2.5 Architecture of the *LeNet-5* CNN . . . 24
2.2.3 Support Vector Machines . . . 26
2.2.3.1 Kernel-SVM . . . 28
2.2.4 AdaBoost . . . 29
2.2.5 Random Forest . . . 30
2.2.6 Unsupervised Machine Learning . . . 31
2.2.6.1 Clustering and Grouping . . . 32
2.2.6.2 Dimensionality Reduction . . . 34
2.2.6.3 Metric MDS . . . 37
2.2.6.4 Kernel-PCA . . . 37
2.2.7 Unsupervised Neural Networks . . . 39
2.2.7.1 Neural-PCA and Neural-ICA . . . 39
2.2.7.2 Self-Organizing Maps and Neural k-means . . . 40
2.2.7.3 Recurrent Neural Networks . . . 42

3 Model-based Spectral Clustering **44**
3.1 Graph-Based Spectral Clustering Algorithms . . . 44
3.2 Related Work on Large Scale Spectral Clustering . . . 47
3.2.1 Nyström and Column-sampling Methods . . . 49
3.2.2 Low-rank Approximations . . . 50
3.3 Modifications and Alternatives to Spectral Methods . . . 51
3.3.1 Independent Component Analysis . . . 53
3.3.2 Combined Approaches . . . 55
3.4 Perceptual Grouping Strategies in Biomedical Imaging . . . 57

3.4.1 Related Work on Perceptual Grouping with Similarity Models 58
3.4.1.1 Intensity and Proximity Models 59
3.4.1.2 Grouping Co-linear Line Segments 60
3.4.1.3 Continuation and Membrane Gap Completion 60
3.4.1.4 Combining Color and Boundary Information 61
3.4.1.5 Grouping Superpixels 61
3.4.1.6 Integrating Texture and Contours 62
3.4.1.7 Learning Affinity Models 63
3.5 Summary of Background Research 64

4 Localization of Mitochondria based on Clustering Extracted Line Segments 67
4.1 Introduction 67
4.2 Feature Extraction Methods 70
4.3 Grouping Objective 72
4.4 Similarity Measures 73
4.4.1 Orthogonal Projections 73
4.4.2 Projection-to-Distance Ratio 74
4.4.3 Similarity Matrix and its Properties 76
4.5 Grouping Strategy 80
4.6 Metric Embedding and kernel-PCA 82
4.7 Experiments on Synthetic Data Sets 82
4.8 Application to Mitochondria Localization 89
4.8.1 Dimensionality Unfolding Principle 92
4.8.2 Results of Single Linkage Hierarchical Clustering 93
4.9 Contour Integration with Rotated Line Segments 95
4.10 Discussion 96

5 Model-based Spectral Clustering Framework for Segmentation of Mitochondria 100
5.1 Introduction 100
5.2 Recursive Spectral Clustering Framework 102
5.3 Proposed Adaptive Similarity Models 104
5.4 Experimental Results 107

6 Anticorrelation-based Dimensionality Reduction 112
6.1 Introduction 112
6.2 Development of the Spectral Clustering Equivalence 114
6.3 Results of Comparison to kernel-PCA 116
6.4 Connection of SCE with Normalized Cuts 117
6.5 SCE Extension with Latent Variables 118

7 Thesis Conclusion and Future Work 121
7.1 Contributions and Conclusions 121
7.2 Future Work 123
7.2.1 Clustering Projected Line Segments 124
7.2.2 Mitochondria Segmentation 125
7.2.3 Anticorrelation-based Dimensionality Reduction 125

Bibliography 127

List of Figures

1.2.1 Main components of mitochondria 3
1.2.2 Challenges in subcellular image segmentation of apoptotic mitochondria 4
1.2.3 Major challenges in subcellular segmentation depend on the tissue and cell type and include varying size and shape of organelles 5
1.2.4 TEM image of a cell showing clustered and deformed mitochondria 6
1.5.1 Diagram of the achieved contributions towards the PhD thesis 10

2.2.1 The Perceptron as a mathematical model of a biological neuron 17
2.2.2 Example of two different Multi Layer Perceptron architectures 19
2.2.3 Example of a deep CNN architecture 22
2.2.4 Input-output characteristics of an S-cell in the mathematical neocognitron model . 23
2.2.5 Architecture of the Convolutional Neural Network LeNet-5 25
2.2.6 The idea of SVM classifiers 26
2.2.7 Mathematical foundations of SVM and kernel-SVM 28
2.2.8 An example of a single linkage hierarchical clustering 33
2.2.9 Neural architectures for nonlinear PCA and linear ICA 40
2.2.10 Self Organizing Maps and Recurrent Neural Networks 41

3.3.1 Generic architecture of the proposed model-based spectral clustering approach to the unsupervised segmentation of mitochondria 52
3.4.1 Selected laws of perceptual organisation 58
3.4.2 Variables involved in the computation of the similarity of two co-linear line segments 60

4.1.1 Mitochondrial morphology as a set of line segments 68
4.2.1 Comparison of TEM images of clustered versus well-separated mitochondria . . . 71
4.3.1 Line grouping objectives 73
4.4.1 Variables involved in the computation of a pairwise projection-based similarity measure for grouping two line segments 74
4.4.2 An example of a set of line segments that violate the symmetry and the triangle inequality metric requirements. 76
4.4.3 Spectral properties of the constructed similarity matrix 78
4.7.1 Different kernels and line clustering results for two synthetic mitochondria 83
4.7.2 Synthetic structures with added stationary noise. Part I 85
4.7.3 Synthetic structures with added stationary noise. Part II 86

4.7.4 Synthetic line structures with added varying noise. Part I 87
4.7.5 Synthetic line structures with added varying noise. Part II 88
4.8.1 Mitochondria localization experiment. Clustering with kernel-PCA and k-means . . 90
4.8.2 Mitochondria localization and clustering results for $k > 2$ and $L \geq 2$ 91
4.8.3 Dimensionality unfolding algorithm 92
4.8.4 Dimensionality unfolding-based results for $L = 2$ dimensions 93
4.8.5 Single linkage hierarchical clustering results for mitochondria images 94
4.9.1 Illustration of the difference between restricted and full similarities 95
4.10.1 Contour integration experiment with rotated line segments 97
4.10.2 Comparison between k-means and SLHC-based localization of mitochondria . . . 99

5.2.1 Architecture of model-based recursive spectral clustering framework for the segmentation of mitochondria 103
5.3.1 Concept of adaptive similarity mixtures 106
5.3.2 Proposed four-stage cascaded raw data normalization and data fusion 107
5.4.1 Ground Truth Mitochondria Segmentation 108
5.4.2 Selected qualitative segmentation results for the ASCB Dataset I 109
5.4.3 Segmentation results on the ASCB Dataset II 110
5.4.4 Segmentation results using semi-automated tracing software *Livewire* 111
5.4.5 Quantitative performance results 111

6.2.1 Description of the SCE algorithm and its comparison to spectral clustering 115
6.3.1 Asymmetric data experiment 116
6.4.1 Comparison between Ncut- and SCE-based segmentation 117
6.5.1 Concept of the Ising-based SCE 119

List of Tables

2.2.1 Perceptron Learning Algorithm 18
2.2.2 Multi Layer Perceptron Learning Algorithm 20
2.2.3 Architectural ideas of the LeNet-5 CNN 24
2.2.4 Parameters of the LeNet-5 Convolutional Neural Network 25
2.2.5 Stopping Criteria for Single Linkage Hierarchical Clustering 32
2.2.6 The k-means Algorithm 34
2.2.7 Key Arguments for Dimensionality Reduction 35
2.2.8 Dimensionality Reduction Algorithms 36

3.1.1 Unnormalized Spectral Clustering Algorithm 45
3.1.2 Normalized Spectral Clustering Algorithm according to Shi and Malik 45
3.1.3 Normalized Spectral Clustering Algorithm according to Ng, Jordan and Weiss . . . 46
3.1.4 Affinity Factorization Algorithm according to Perona and Freeman 46
3.3.1 Combined Dimensionality Reduction Algorithms 55
3.4.1 Summary of Reviewed Similarity Models 63
3.5.1 Summary of Background Research on Cellular and Subcellular Segmentation . . . 64
3.5.2 Statistics of the Reviewed Segmentation Methods 65

4.4.1 Supplementary Data for the Set of Line Segments that Violate Metricity. 75
4.4.2 Notations used in Chapter 4 76
4.4.3 Dissimilarity Mapping for the Set of Line Segments. 79
4.5.1 Comparison of two selected grouping strategies 81

5.1.1 Notations used to describe the adaptive similarity models and the recursive spectral clustering framework for the segmentation of mitochondria 102

6.5.1 Proposed Ising-based SCE extension 118

List of Abbreviations

ART	Adaptive Resonance Theory
BSS	Blind Source Separation
CCA	Canonical Correlation Analysis
CDA	Covariance Discriminant Analysis
CNN	Convolutional Neural Network
CSE	Constant Shift Embedding
EM	Electron Microscopy
ET	Electron Tomography
GLCM	Gray Level Co-occurance Matrix
HOG	Histograms of Oriented Gradients
ICA	Independent Component Analysis
KICA	Kernel Independent Component Analysis
LBP	Local Binary Patterns
LDA	Linear Discriminant Analysis
LLE	Locally Linear Embedding
MDS	Multidimensional Scaling
MLP	Multi Layer Perceptron
NMDS	Nonmetric Multidimensional Scaling
NN	Neural Network
NPCA	Neural Principal Component Analysis
OCR	Optical Character Recognition
PCA	Principal Component Analysis
PSD	Positive Semidefinite
PDS	Positive Definite Symmetric
RNN	Recurrent Neural Network
SCE	Spectral Clustering Equivalence
SEM	Scanning Electron Microscopy
SLHC	Single Linkage Hierarchical Clustering
SOM	Self-Organizing Map
ssTEM	Serial Section Transmission Electron Microscopy
SVD	Singular Value Decomposition
SVM	Support Vector Machines
SPSD	Symmetric Positive Semidefinite
TEM	Transmission Electron Microscopy

Abstract

A Machine Learning Approach to the Unsupervised Segmentation of Mitochondria in Subcellular Electron Microscopy Data

Julia Dietlmeier

Recent advances in cellular and subcellular microscopy demonstrated its potential towards unravelling the mechanisms of various diseases at the molecular level. The biggest challenge in both human- and computer-based visual analysis of micrographs is the variety of nanostructures and mitochondrial morphologies. The state-of-the-art is, however, dominated by supervised manual data annotation and early attempts to automate the segmentation process were based on supervised machine learning techniques which require large datasets for training. Given a minimal number of training sequences or none at all, unsupervised machine learning formulations, such as spectral dimensionality reduction, are known to be superior in detecting salient image structures. This thesis presents three major contributions developed around the spectral clustering framework which is proven to capture perceptual organization features. Firstly, we approach the problem of mitochondria localization. We propose a novel grouping method for the extracted line segments which describes the normal mitochondrial morphology. Experimental findings show that the clusters obtained successfully model the inner mitochondrial membrane folding and therefore can be used as markers for the subsequent segmentation approaches. Secondly, we developed an unsupervised mitochondria segmentation framework. This method follows the evolutional ability of human vision to extrapolate salient membrane structures in a micrograph. Furthermore, we designed robust non-parametric similarity models according to Gestaltic laws of visual segregation. Experiments demonstrate that such models automatically adapt to the statistical structure of the biological domain and return optimal performance in pixel classification tasks under the wide variety of distributional assumptions. The last major contribution addresses the computational complexity of spectral clustering. Here, we introduced a new anticorrelation-based spectral clustering formulation with the objective to improve both: speed and quality of segmentation. The experimental findings showed the applicability of our dimensionality reduction algorithm to very large scale problems as well as asymmetric, dense and non-Euclidean datasets.

Chapter 1

Introduction

1.1 Subcellular Imaging and Electron Microscopy

Electron microscopy has emerged as a powerful technique to address fundamental questions in molecular and cellular biology [66]. It makes possible visualization of the molecular architecture of complex viruses, organelles and cells at a resolution of a few nanometres. In the last decade EM has allowed major breakthroughs that have provided exciting insights into a wide range of biological processes. In particular, in electron tomography the biological sample is imaged with an electron microscope, and a series of images is taken from the sample at different views. Transmission electron microscopy provides resolutions in the order of nanometer. Hence it is a critical imaging modality for biomedical analysis at the sub-cellular level. Prior to imaging, the sample has to be specially prepared to withstand the conditions within the microscope. Subsequently, those images are processed and combined to yield the three-dimensional reconstruction or tomogram. Afterwards, a number of computational steps are necessary to facilitate the interpretation of the tomogram, such as noise reduction, segmentation and analysis of subvolumes. As the computational demands are huge in some of the stages, high performance computing techniques are, for example, used to make the problem affordable in reasonable time [66].

Electron tomography has got the ability to visualize in three dimensions the subcellular architecture and macromolecular organization of native cells and tissues at nanometer scale. This resolution level allows the study of the 3D organization of the structural components at a detail sufficient for the identification of macromolecular complexes, the quantitative analysis of their abundance, and their spatial distribution as well as their interactions in the native cellular context.

One area of rapid development is volume electron microscopy, a collective term for EM techniques focusing on analysis of "large" volumes ("large" being a relative description in the EM world). Most techniques that fall under the volume EM umbrella were initially developed for examination of the central nervous system [163], a result of the need to analyze axons and dendrites that span large distances with sufficient resolution to detect individual synaptic vesicles and densities. The associated technical challenge lay in overcoming the "field of view versus resolution" problem, to enable visualization of a single specimen across different scales, a problem common to most imaging modalities.

Volume EM can be performed using transmission or scanning electron microscopes. Each approach has its own strengths and weaknesses, and the choice is dependant on the required lateral (x, y) and axial (z) resolution, and the size of the structure of interest. Historically, transmission electron microscopy was the tool of choice for ultrastructural examination of biomedical specimens at sub-nanometer resolution. However, for many cell biology studies structural resolution is actually limited by the deposition of heavy metals onto membranes during sample preparation. In addition, voxel dimensions may only need to be half that of the smallest expected feature of interest. Advances in Scanning Electron Microscopy technology are now driving a paradigm shift in electron imaging. SEMs with field emission electron sources and high efficiency electron detectors can achieve lateral resolutions in the order of 3nm, allowing visualization of structures such as synaptic vesicles and membranes though resolving individual leaflets of membrane.

Analysis of electron microscopy images is normally carried out by specialists with experience in the identification and interpretation of biological features in the complex grayscale world of electrons. However, manual analysis is labor intensive and can be excruciatingly slow. It is widely acknowledged that more automation in the downstream processing pipeline is essential for accelerating segmentation and structural analysis [163, 26, 47, 91, 97, 106]. Despite the existence of computational methods, none has stood out as a general applicable method yet, and manual segmentation still remains the prevalent method . Progress has been made in this area, and in the development of more sophisticated tools, but these approaches remain limited to defined structures such as for example mitochondria [76, 50, 79, 132, 151] and synapses segmentation [111].

1.2 Challenges in Mitochondria Segmentation and Apoptosis

The field of nano-biophotonics and imaging of subcellular regions, in particular of mitochondria, is an extremely complex and dynamic environment. Mitochondria form an important category of membrane enclosed *organelles* which reside inside every living cell. Mitochondria have an average diameter of 200nm with huge variation in size and shape even within one cell, which are likely to move within a living cell and undergo fission and fusion. Mitochondrial morphology depends on the type of biological tissue and further undergoes structural changes during induced or naturally occurring biochemical processes [199]. This fact accounts for the vast range of mitochondrial shapes and textures and challenges a unified approach to localization and segmentation.

Mitochondria consist of two major compartments: the intermembrane space and the matrix as can be seen in Fig. 1.2.1. The outer membrane engulfs an inner membrane, which is folded into *christae* which is clearly visible in the electron microscope images. Mitochondria play an important role in the processing of the food molecules into adenosine triphosphate (ATP) which provides energy for the cells, as well as in several cellular functions such as signaling, differentiations, cell growth and mitochondrial regulation processes. In addition to their role in cellular bioenergetics, mitochondria also initiate common forms of *programmed* cell death (apoptosis) through the release of proteins such as cytochrome *c* from the intermembrane and intracristal spaces [199]. Apoptosis is an evolutionary conserved cell death process that is fundamental to remove superfluous and damaged cells from the bodies of multicellular organisms. Individual cells within a population undergo apoptosis at distinct, apparently random time points [169].

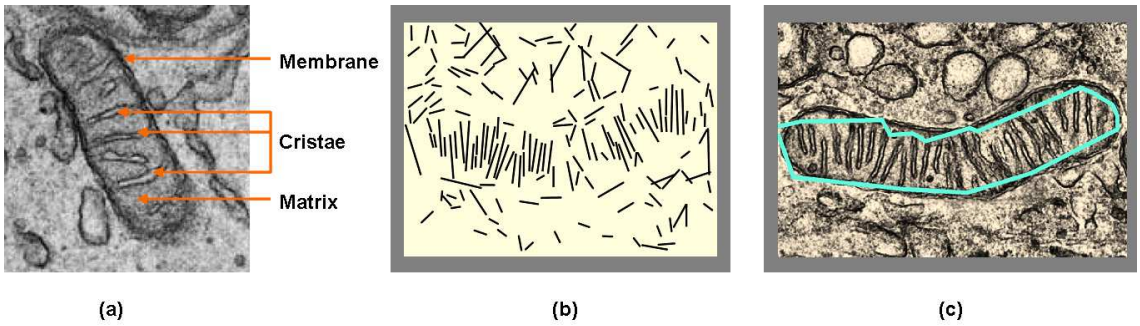


Figure 1.2.1: (a) EM image of a mitochondrion with its main components. (b) Set of extracted line segments. (c) Localization hull overlayed on the original image. The method deployed for the grouping of line segments describing mitochondrial morphology is presented in Chapter 4.

Cell death in general is an essential phenomenon in normal development and homeostasis, but also plays a crucial role in various pathologies [67]. Specifically, alterations in apoptotic pathways result in a loss of the balance between cell proliferation and cell death, leading to a number of diseases in humans. For example, abnormal down-regulation of apoptosis is an important contributor to cancer and autoimmune diseases, whereas excessive up-regulation of cell death is implicated in neurodegenerative disorders such as Alzheimer's, Parkinson's and Huntington's [191, 100]. Currently, there is overwhelming and widely-accepted evidence of impaired mitochondrial function as a causative factor in these diseases.

Our special interest in apoptosis is connected to the research by our biomedical partners at the Royal College of Surgeons in Ireland (RCSI), Department of Physiology and Medical Physics. The researchers are performing experiments on human prostate cells DU-145 treated with the apoptosis inducer staurosporine (STS) as well as HeLa cells treated with tumour necrosis factor related apoptosis inducing ligand (TRAIL) [169]. High-speed subcellular imaging of the cell interior and mitochondrial outer membrane permeabilisation is being used to analyze cellular mitotic history, study apoptotic signaling and the mitochondrial response to apoptosis inducing agents.

As has been shown by Sun et al. [199], mitochondria undergo morphological changes during apoptosis. Especially the change of the intramitochondrial christae structure has been reported and is likely to coincide with the permeabilisation of the outer mitochondrial membrane. In order to examine the ultrastructure of mitochondria (present in HeLa cells treated with *etoposide*) at defined stages of apoptosis in an asynchronous cell population, fluorescence microscopy was first used to characterize the state of apoptosis. Then correlated light and three-dimensional electron microscope tomography was performed on cells growing in special petri dishes that contain a glass coverslip with an etched grid in order to study the sequence of structural changes. As a result, the authors have identified five characteristic mitochondrial morphologies based on observation of electron micrographs of apoptotic HeLa cells as shown in Fig.1.2.2: *normal*, *normal-vesicular*, *vesicular*, *vesicular-swollen* and *swollen*.

The process of segmenting the interior of a cell composed of organelles and cytosol is confronted by a variety of challenges which according to their complexity can be mainly categorized into minor and major factors. There are minor problems which originate from processing low contrast and noise degraded images acquired under inadequate lighting conditions.

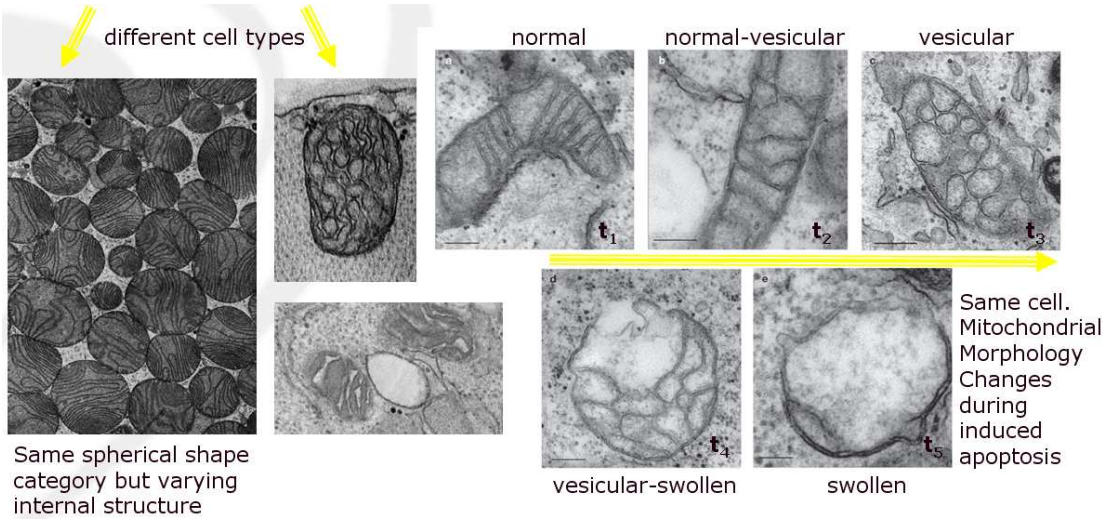


Figure 1.2.2: Challenges in subcellular segmentation of *apoptotic* (here: experimentally induced programmed cell death process in HeLa cells [199]) mitochondria which exhibit time-varying morphologies (t_1 to t_5). These substantial changes from the *normal* (t_1) to the eventually *swollen* (t_5) morphology have been detected through joint correlated three-dimensional light and electron microscopy imaging. Image source: NBIP Ireland ITC Workshop [53].

Thus, in low quality TEM images the degradation can be due to speckle noise and non-uniform illumination. Specifically, the contrast of images in electron microscopy is determined by the nature and extent of interactions between the electron beam and the specimen. Therefore, the resolution available for untreated biological samples is limited by the physical constraints on EM. Properties of both the specimen (inherent contrast) and of the microscope system (instrumental contrast) are of importance. The level of contrast is determined by the average atomic number of the specimen, and biological specimens consist mainly of carbon, nitrogen, oxygen, and hydrogens atoms. Because EM is imaging the differences in scattering between low atomic number components such as carbon (cells) and oxygen (water), the contrast is often low and may depend strongly on the local concentration of heavier ions [204]. Thus, the weak contrast is a limiting problem in the imaging of biological specimens. The contrast is generally enhanced by (i) directly increasing the specimen contrast using various preparation procedures (i.e. staining, shadowing, etc.) or (ii) using longer exposure times in recording the image. The image processing techniques in enhancing contrast may include such standard methods as gray level histogram modification.

Nonuniform intensity distribution is another problem associated with TEM images and leads to variations in brightness due to electron imaging defects or non-uniform support films and specimen staining. These variations render image processing operations such as segmentation more difficult. Some correction techniques require estimation of the global illumination field [202]. Intensity nonuniformity is also referred to as *intensity inhomogeneity*, *shading* or *bias field*. A shading correction can also be done in the following way. Firstly, the original image is smoothed by applying a few times a low-pass filter of a large size. Secondly, the difference between the original and the smoothed images is calculated and the resulting image is autoscaled.

The presence of noise in electron microscopy images can be generally described as the random variation in the pixel content caused by the acquisition, digitization and transmission processes.

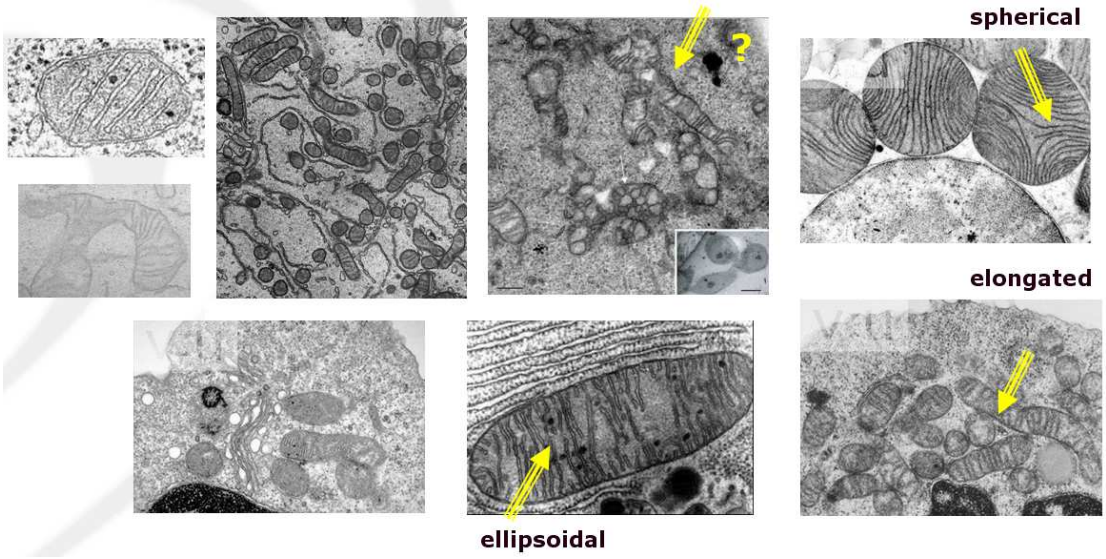


Figure 1.2.3: Major challenges in subcellular segmentation depend on the tissue and cell type and include varying size and shape of organelles. Image source: NBIP Ireland ITC Workshop [53].

Microscopy applications often have to contend with low light conditions. This can be, for example, a case when samples are stained with fluorescent dyes and thus emit very little light. Dyes, which emit light when irradiated, are used as biomarkers in modern microscopy. Because many dyes are phototoxic, providing more light to the cell to receive more signals from the biomarker, can potentially destroy the cell. Therefore, the light exposure is kept as low as possible, leading to the camera noise which is strongest at low light levels and often has detrimental effect on dark images. Depending on the application, users have to choose a suitable camera in the tradeoff between high sensitivity or low noise. However, hardware noise can not be avoided altogether, but it can be reduced by a process called smoothing or filtering. Often, smoothing by averaging or median filtering is applied. Low-pass filters are also applied for image smoothing. Their application is based on the discrete convolution of the original image with a special mask. The effect of the convolution depends on the type of filter kernel used. One way to make an image sharper is to apply edge detection filters such as the Laplace, Roberts or Sobel filters and to add the found edges to the original electron microscopy image.

Major challenges in the analysis of subcellular micrographs, however, come from the complexity of the dynamic intracellular environment and include different sizes and shapes of organelles and varying internal structure of mitochondria due to the inner membrane folding. That is, different living organisms, different tissues and cell types have all distinct and unique mitochondrial morphologies, as can be seen in Fig. 1.2.3. In addition, in studies on human cells, apoptotic organelles have shown to exhibit time-varying changes in shape and texture. Because a cell is in a fact a three-dimensional object, the standard sample preparation in electron microscopy includes the freeze and subsequent multiple mechanical slices through the cell. After that process, each slice is imaged and the volumetric information can be acquired afterwards. The image artifacts associated with the sample preparation can include, for example, traces of staining. Other artifacts include signs of slicing and (most profoundly difficult to handle) incomplete shapes of organelles.

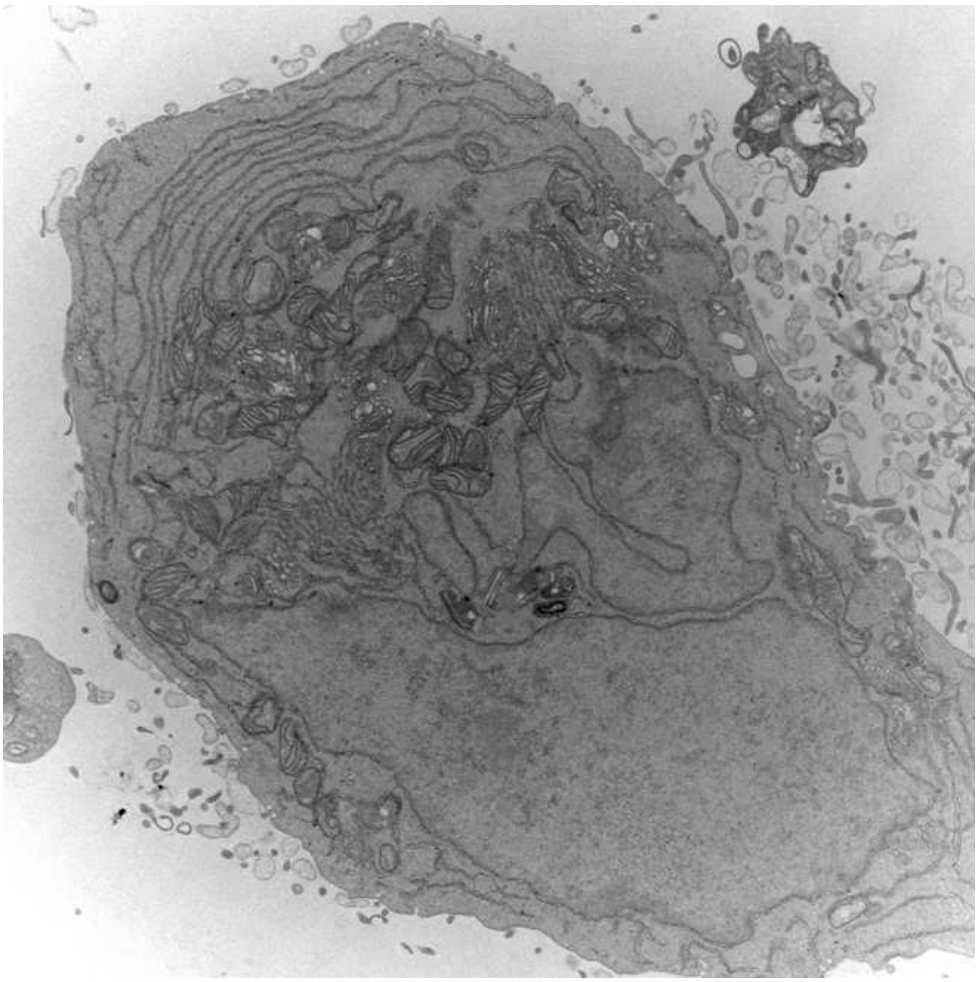


Figure 1.2.4: TEM image of a whole DU-145 human prostate cell showing clustered and deformed mitochondria even before the apoptosis induced morphological changes. Image source: RCSI.

Staining in general, however, needs not to be a problem. Substantial advantages in imaging can be achieved by staining the cells with high atomic number markers such as electrondense stains (heavy metal staining). Differential staining of organelles inside cells using electrondense materials is a standard in EM technique. High resolution and specificity of detection can be also achieved by binding of antibodies or other ligands such as nanoparticle gold labels [204]. Therefore, staining and labeling on demand can indeed provide valuable markers for the subsequent image analysis.

This work looks specifically at the segmentation of mitochondria which vary widely in size, shape and internal structure depending on the tissue and cell type, such as depicted in Fig. 1.2.3 and Fig. 1.2.2. Other significant challenges arise from the presence of clustered and attached mitochondria and cluttered background as shown in Fig. 1.2.4. In many applications, cells, mitochondria and other nanostructures are frequently neighboring or overlapping on each other, which makes segmentation and quantification difficult. In addition, the scale and the resolution of the acquired images have to be taken into account. Overall, the combined challenges determine the segmentation strategy and applied methods and require the development of advanced techniques. Currently, there is no unified approach for different acquired micrographs and the research work on cellular and subcellular segmentation is mostly divergent and application specific [219].

TEM images of mitochondria are able to capture the intrinsic structural elements that are caused by the inner membrane folding. This evidence suggests the feasibility of a feature-driven segmentation approach. In this work, we present a multi-stage segmentation approach composed of preprocessing, object registration and shape extraction. The preprocessing step is composed of contrast enhancement, structural noise removal and the compensation for the uneven illumination. Also, in order to enhance the appearance of the inner membrane folding, the non-linear and shape preserving feature enhancement techniques are considered. The next step is to determine the location of the mitochondrion (seed point/marker) and to extract its boundaries. Localization is based on the a *priori* assumption of mitochondrial morphology of interest and foresees the extraction and grouping of low-level membrane primitives.

1.3 Advances in Cellular Image Segmentation

There are many different methods that were proposed and claimed to be superior in segmenting some specific type of cells, mitochondria and other nanostructures. Some of these methods are based on watershed segmentation, region growing or neural networks. Other include Bayesian methods, linear discriminant preprocessing, tabu search, correlation clustering [229], multilayered segmentation, membrane pattern method, iterative feedback and machine learning [219]. A morphological method that enables automated morphology analysis of partially overlapping nanoparticles in electron micrographs has been presented in [162]. This method adopted a two-stage approach: the first stage executed the task of particle separation, and the second stage conducted simultaneously the tasks of contour inference and shape classification. Graph cuts-based approaches were also used, for example, by Al-Kofahi et al. [3] to segment cell nuclei in histopathology images. The authors concluded that the confounding image characteristics that led to most detection/segmentation errors were high cell density, high degree of clustering, poor image contrast and noisy background, damaged/irregular nuclei, and poor edge information. Kaynig et al. [103] addressed the problem of neuron geometry extraction by perceptual grouping in ssTEM images. The architecture comprised a Random Forest for classifying single pixels, and novel energy terms for membrane segmentation with graph cut optimization. However, none of the above methods can meet the general purpose for so many different types of cellular and subcellular structures.

In order to overcome at least some of these drawbacks, authors in [219] proposed, for example, a more generalized approach. This method utilized the general property of the cell and nanoparticle images: intensity and intensity gradient and thus was suitable for different types of cell and nanoparticle images. Martinez-Sanchez et al. [141] targeted the segmentation of membranes in electron tomography. This method was based on local differential structure and on a Gaussian-like membrane model. Detection of membranes plays an important role in segmentation as they encompass compartments within biological specimens, define the limits of the intracellular organelles, cells and the inner membrane folding of mitochondria. Ghita et al. [76] used the concept of *superpixels*[1] to segment membranes in high-resolution EM images. Also, the concept of superpixels has been applied in [132, 133] and [229] to reduce the initial complexity of EM data

[1] A superpixel is an image patch which is better aligned with intensity edges than a rectangular patch.

prior to the graph-based segmentation. More discussion on related works and methods on cellular image segmentation is provided in Section 2.1. The summary of background research is given in Section 3.5. Our review show that the requirement for robust segmentation and quantification of cells, mitochondria and other nanostructures increases significantly due to the rapid development of microscopy imaging technology. It still remains challenging due to the diversity of cell or nanoparticle types, the arbitrary shapes, and the large numbers of cells or mitochondria. To conclude, most existing methods are only capable of segmenting some specific types of cells, organelles and other nanostructures. Most methods are diverging rather than converging to a robust and unified solution and the performance and applicability of these methods thus remain limited.

A promising approach to the diversity challenge of cellular segmentation and classification is based on supervised machine learning with convolutional neural networks. For instance, Ronnenberber et al. [172] won the ISBI cell tracking challenge 2015 and developed a new convolutional neural network for biomedical image segmentation. This winning deep-learning architecture can work with very few training images and has been successfully tested on neuronal structures in EM stacks as well as on cell segmentation in light microscopy images.

1.4 Overview of Objectives and Contributions

Deep learning with convolutional neural networks and machine learning in general are currently two popular topics in the computer vision community [161]. These methods gained their deserved recognition due to their exceptional results in image segmentation quality in many applications including also biomedical imaging.

While supervised machine learning frameworks foresee large training sets consisting of annotated ground truth data, the guiding idea for this work is to use the unsupervised machine learning and model-based approach which can be combined with the spectral clustering realization to produce desired segmentation results. The reason for this choice is two fold. Initially, we have been limited by the number of images provided by our biomedical partners. Therefore we had to discard the idea of using the supervised or semi-supervised approaches which require large databases of manually-annotated images needed to train classifiers. Then, we have observed that in the images provided, the localization and segmentation of mitochondria can be cast within the framework of perceptual organization which is the key pre-attentive feature of the evolutional human vision.

In this thesis, the chosen framework of spectral clustering combines both arguments: it belongs to the *unsupervised* machine learning class of algorithms, and it is able to integrate different perceptual grouping cues into a joint similarity (*affinity*) model. If the latter has been designed properly and according to the laws of visual segregation, the resulted foreground background segmentation in the spectrally induced feature space is known to capture the perceptual organization in images or networks of co-linear line segments. That is, the machine *learning* in this context is associated with the *unsupervised* discovery of structures (clusters) in the input data, and the modeling, dimensionality reduction and analysis of the underlying manifold. A special interest in spectral clustering is also based on its known ability to discover significant relations which otherwise can not be detected in the input data [220]. Additionally, the stability of spectral clustering algorithms is well understood and the framework of noise analysis and reduction is well defined [176].

For the particular type of mitochondria provided a set of descriptive features can be inferred and defined. These can rely on the low-level primitives such as shape, contours and line segments, and therefore we may for example discriminate between spherical, elliptical or elongated structures. Additionally, the textural appearance of the inner membrane folding can serve as a grouping cue. The set of descriptive features is used to design a similarity model which then serves as input to the selected spectral clustering algorithm and results in the unsupervised classification.

Therefore, the first main contribution of this research targets the localization of mitochondria of *lamellar* or *tubular* morphology in electron microscopy images. Results of the research on this topic are presented in Chapter 4. In particular, mitochondria of *lamellar* and *tubular* morphology are characterized by the linear or quasi-linear inner membrane folding which can be represented by low-level primitives such as line segments. Line extraction results in a network of line segments which can be grouped to infer the original mitochondrial structure. However, the only similarity criterion which is able to distinguish the significant and thus non-accidental collection of these non-parallel line segments in this case is based on the amount of orthogonal projection spanned between the line segments. As there has not been any reported prior work on the similarity models or line clustering involving projections, the first contribution constructs and analyzes a pairwise similarity model which mainly integrates grouping cues based on orthogonal projections and the Euclidean distance between two line segments. The localization of mitochondria provides us with a marker or a seed point which can further be used to extract the outer boundary of mitochondria with for example region growing approaches.

The second main contribution to the segmentation of mitochondria considers the extension of line segments grouping to the case of grouping pixels with the goal of extracting salient membrane structures of subcellular organisms in gray scale electron microscopy images. For this case the similarity model designed combines such variables as intensity (brightness cue) and coordinates (proximity cue) which are measured in different quantities and are contained within different ranges. The research on this topic is presented in Chapter 5 and shows that the normalization of raw data is important for the performance of classification methods. Furthermore, we introduce a novel ratio-based similarity model which considers the proper normalization of raw data and sampling adjustment of the nominator and denominator-based probability density functions. The major challenge encountered was that for the size of an input image $N \times M$, the resulting similarity matrix acquires the size $NM \times NM$. This makes direct spectral decomposition impractical. We solve this problem by subsampling and tiling the input image. However, we asked the question if there can be a low-cost solution to this computational complexity problem of spectral clustering which could produce equivalent classification results.

Therefore, the third main contribution (see Chapter 6) addresses the current limitation of spectral clustering related to its computational complexity and further questions the optimality of eigenvectors. We build on the idea that saliency can be associated with the presence of significant and non-accidental relationships contained in the constructed similarity matrix. We observe that highly anti-correlated columns of the centralized Gram matrices carry discriminative information equivalent to that of eigenvectors. Specifically for the class of non-metric or asymmetric data which results in a generally indefinite similarity matrix, the latter has to be firstly symmetrized and then embedded into the Euclidean space in order to guarantee the applicability of clustering algorithms

which are based on Euclidean geometry. There is a number of works specifically targeting the very large scale limitation. Selected practical implementations therein include parallel and distributed computing, image subsampling or tiling, subsampling of a similarity matrix, exploiting matrix sparsity and estimating eigenvalues and eigenvectors. Therefore, motivated by these limitations we develop a new anticorrelation-based spectral clustering formulation which can be applied to asymmetric, symmetric but non-Euclidean, and dense datasets.

1.5 Publications Emerging from this Research

This thesis is set to provide a solution to recently dominating manual segmentation of mitochondria. The contributions achieved towards final developed image segmentation, mitochondria localization and classification methods are outlined in Fig. 1.5.1.

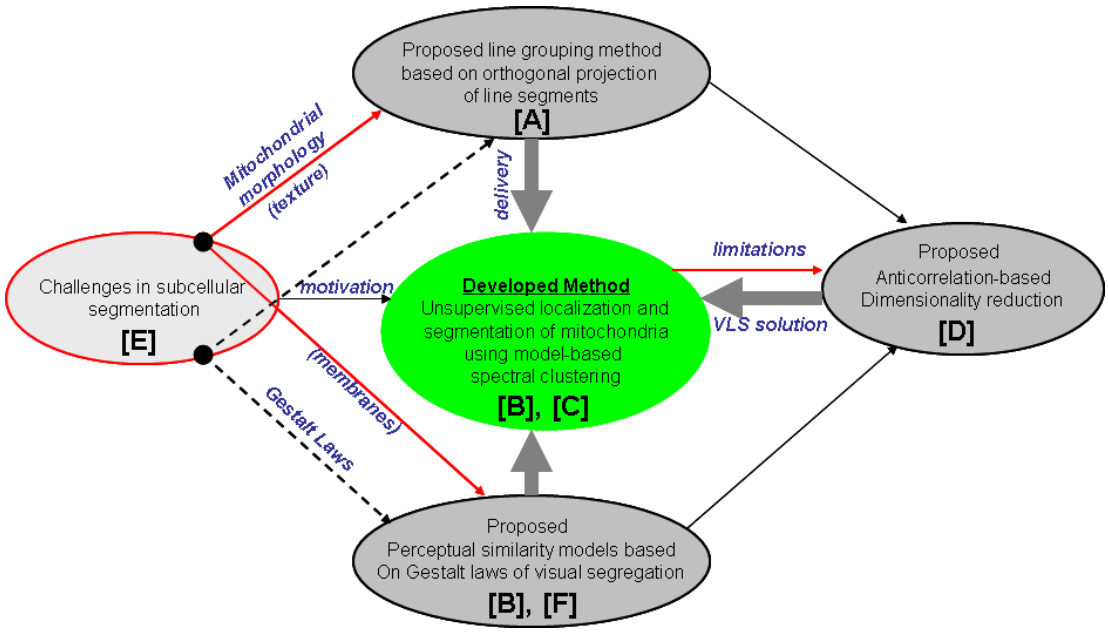


Figure 1.5.1: Diagram of the achieved contributions towards the PhD thesis. This diagram also shows the relationship between the challenges in subcellular segmentation, motivation factors, limitations of the chosen spectral clustering framework and the developed methods.

Journals

[A]: **Pattern Recognition Letters** (published in Elsevier 51, pp:50-56, January 2015). "On the projection similarity in line grouping" by *Julia Dietlmeier*, *Ovidiu Ghita* and *Paul F. Whelan* [51].

[B]: **IEEE Transactions on Image Processing** (published in IEEE Xplore 23(10), pp:4576-4586, 2014). "Automatic segmentation of mitochondria in EM data using pairwise affinity factorization and graph-based contour searching" by *Ovidiu Ghita*, *Julia Dietlmeier* and *Paul F. Whelan* [76].

[C]: **Journal of Structural Biology** (published in Elsevier 184(3), pp:401-408, December 2013). "Unsupervised mitochondria segmentation using recursive spectral clustering and adaptive similarity models" by *Julia Dietlmeier*, *Ovidiu Ghita*, *Heiko Duessmann*, *Jochen H.M. Prehn* and *Paul F. Whelan* [50].

[D]: **Lecture Notes in Computer Science** (published by Springer in LNCS 6915, pp:139-149, 2011). "A new anticorrelation-based spectral clustering formulation" by *Julia Dietlmeier*, *Ovidiu Ghita* and *Paul F. Whelan* [52].

Conferences, Posters, Presentations

[E]: **National Biophotonics and Imaging Platform Ireland ITC Workshop** (NBIPI ITC, presented in 2009). "On the segmentation of intracellular subspaces: challenges and methods" by *Julia Dietlmeier* and *Paul F. Whelan*

[F]: **Biophotonics and Imaging Conference** (BioPic, presented in 2010). "Perceptual Grouping Strategies in Molecular Imaging" by *Julia Dietlmeier*, *Ovidiu Ghita* and *Paul F. Whelan*. **Runner-UP best poster award in postdoctoral category**.

[G]: **The International Machine Vision and Imaging Conference** (IMVIP, **peer-reviewed**, presented in 2008 and **published** in IEEE Xplore). "Cost-Effective HPC Clustering For Computer Vision Applications" by *Julia Dietlmeier*, *Seán Begley* and *Paul F. Whelan*.

[H]: **Biophotonics and Imaging Graduate Summer School** (BIGSS, Presented in 2008). "Automated Segmentation of Mitochondria in EM Images" by *Julia Dietlmeier* and *Paul F. Whelan*.

[J]: **Bioengineering UK Conference** (Oxford, presented in 2009). "Computer-Guided Recognition of Mitochondria in Densely Cluttered Subcellular Environments" by *Julia Dietlmeier* and *Paul F. Whelan*.

[K]: **European Molecular Imaging Meeting** (EMIM, presented in 2010). "Structural Methods in Subcellular Imaging" by *Julia Dietlmeier* and *Paul F. Whelan*.

[L]: **The Annual Conference of the Bioengineering Section of the Royal Academy of Medicine in Ireland** (Bini, presented in 2011). "Towards Unsupervised Segmentation in High-Resolution Medical Nano-Imaging" by *Julia Dietlmeier*, *Ovidiu Ghita* and *Paul F. Whelan*.

[M]: **Biophotonics and Imaging Conference** (BioPic, presented in 2013). "Unsupervised segmentation of mitochondria using model-based spectral clustering" by *Julia Dietlmeier*, *Ovidiu Ghita* and *Paul F. Whelan*.

Chapter 2

Background Research

The motivation for this research comprises of three interconnected factors: (i) High demand for unsupervised segmentation solutions in molecular imaging; (ii) The established performance and suitability of spectral clustering as a machine learning tool in the unsupervised classification and segmentation tasks; (iii) The inherent limitations of spectral clustering regarding its computational complexity. In fact, the application-specific problem (i) cannot be solved without targeting the computational challenge in (iii) and the understanding of the theoretical foundation of spectral clustering in (ii). On the other hand, (ii) and (iii) have been originally motivated by (i) and are set to deliver the solution to the problem in (i). All these factors determine the layout of this chapter.

Therefore, we organize this chapter as follows: in Section 2.1 we review the state of the art works on molecular and mitochondria segmentation tools. The machine learning field consists mainly of two major areas: supervised and unsupervised implementations and there are many algorithms which combine both. First, the most commonly used supervised learning algorithms and the theory behind are reviewed in detail in Section 2.2.1. Therein, the currently important category of convolutional neural networks and its relation to image segmentation is discussed in Section 2.2.2.3. Second, unsupervised machine learning has been narrowed to the clustering and spectral dimensionality reduction algorithms which are discussed in Section 2.2.6. Spectral clustering itself is a special combination of both dimensionality reduction and clustering in the projected low-dimensional manifold. We then proceed with the analysis of the family of conventional graph-based spectral clustering algorithms in Section 3.1.

As we have already mentioned, our interest in very large scale spectral clustering has originated from the necessity to solve pixel classification problems. As we have arrived at computational limits of spectral clustering we cannot process large images without resorting to subsampling, block processing or approximation techniques. Therefore Section 3.2 reviews the relevant work on the numerical solutions to very large scale spectral clustering, while its modifications and alternatives are outlined in Section 3.3. We point to the number of publications on spectral clustering and its efficient implementations. These papers indicate that this research field is still extremely active.

The quality of spectral image segmentation or input domain partitioning in general (as we also want to cluster line segments which describe mitochondrial morphology) is directly related to the cue integration and mathematical formulations of similarity measures. The latter often contain application-dependent and non-linear kernel functions in order to allow the linear separability of

clusters in the feature space. The important notion of kernels is first introduced in Section 2.2.3.1 and appears in different formulations such as, for example, kernel-SVM, kernel-PCA or kernel-ICA throughout this chapter. In our case we also derive similarity models based on perceptual grouping principles. Therefore, the last Section 3.4.1 briefly discusses the organizational laws of visual perception, its applicability to molecular imaging and presents related works on similarity models which integrate perceptual grouping cues.

2.1 Subcellular Segmentation

A recent review on segmentation tools in electron tomography [12] emphasized the need for more objective segmentation tools. Manual segmentation is the current state of the art [165], and to illustrate how time-consuming this process is, we refer to the work by Marsh et al. published in [139]. The authors indicate that image acquisition and the interpretation of the complete Golgi ribbon required approximately nine months of manual segmentation followed by approximately three months of additional editing for detailed analysis.

The authors in [158] reported a throughput improvement as compared to the previous study and remark that by using a manual segmentation software package IMOD [110] they were able to mark-up all of the mitochondria in less than 15 hours. In their study they demonstrated that an enormous volume of Electron Tomography (ET) data can be acquired and reconstructed relatively quickly in two to three weeks. However, they concluded that the extraction of spatial and quantitative information from such data still remains relatively slow.

Perkins et al. [165] also used the IMOD software package for segmentation of mitochondria. In their work, the authors concluded that this step is rate-limiting in ET throughput, where usually more than 100 slices are present in the reconstruction and often more than 30 objects are segmented. The existing semi-automatic solutions are primarily data-driven and therefore constrained by the tissue type examined and the morphology of mitochondria or other cellular structures. For example, the authors in [85] addressed the detection and segmentation of mitochondria in human skeletal muscles. Mitochondria in these cells are known to have spherical or elliptical shape and uniform texture. In this study the mitochondrial delineation is automated but still requires the selection of an initial seed point inside the organelle.

The main hypothesis employed in the development of computer vision solutions for mitochondria segmentation is based on the observation that these sub-cellular organelles are defined by closed structures with distinct inner textures. Thus, two main directions of research emerged that attempted to exploit specific knowledge in relation to the morphology and structural characteristics of mitochondria [50]. In this regard, a good example of this approach is represented by the work of Narasimha et al. [153] where they propose a texton-based algorithm for mitochondria detection which involves the joint classification using k-Nearest Neighbor, Support Vector Machines and adaptive boosting (AdaBoost). This approach has been applied to the segmentation of mitochondria in MNT-1 cells and the authors conclude that their method performed similar and in some situations better than semi-automatic techniques based on level sets.

A similar idea was followed in [216] where the authors employed the standard Gabor filtering method in conjunction with a Gentle-Boost classifier to detect mitochondria in rat brain tissues.

The experiments reported indicated relative large misclassifications (false acceptance rate is 25% and false rejection rate is 20%), and this fact highlights two major disadvantages associated with the texture-based methods: the complexity of the training process and the fact that these approaches are poorly equipped to adapt to changes in the cristae structures that are induced by cellular and mitochondria membrane events [87].

To counteract the limitations associated with structure-based segmentation methods when applied to mitochondria segmentation, alternative approaches based on active contours [151, 79] and graph partitioning [133, 132, 205] have been proposed. Active contours methods in particular proved successful when deployed in the implementation of semi-supervised segmentation algorithms, but they have shown to be impractical when applied in conventional forms to mitochondria EM data. This was due to several issues such as difficulty in obtaining accurate contour initialization, weak gradients caused by low intensity profiles that are often characteristic for mitochondria membranes, random textures, noise, and more importantly difficulties in deriving statistical models that describe the modes of variation of the mitochondria shapes.

To mitigate these limitations, Seyedhosseini et al. [184] combined the use of algebraic curves and texture to identify image patches that resemble mitochondria which are later ranked by a Random Forest classifier. They have tested their algorithm on mouse neuropil and Drosophila VNC (ventral nerv cord) data and they reported promising results. In an effort to redress the problems associated with traditional level sets implementations when used for segmentation of sub-cellular structures, Nguyen and Ji [156] initially embedded the watershed segmentation into an energy minimization framework (watersnake) and then they incorporated prior information in the form of fixed and variable shape terms that constrain the space that is spanned during the contour propagation process. They quantitatively evaluated their algorithm on rat liver mitochondria data and they demonstrated that the inclusion of prior shape information proved a key element in achieving accurate segmentation results. In spite of these performance improvements, the use of active contours for mitochondria segmentation proved problematic due to contour initialization errors and a large set of parameters that require optimization. Therefore, these approaches were successful when applied to cellular data where the range of mitochondria shapes varies within a restricted domain.

The restricted domain limitation was recently addressed by the application of graph partitioning algorithms to mitochondria segmentation, and the guiding idea was to identify the cycles corresponding to mitochondria contours in undirected graphs. This is usually obtained by enforcing shape minimization constraints to an undirected graph where the nodes are defined in more simplistic cases by pixels, or in more involved approaches by adjacent regions resulting from a pre-segmentation step. Building on this idea, Lucchi et al. [133, 132] combined a primary segmentation step that involves the calculation of superpixels with a graph cut algorithm where unary and pairwise potentials of an energy function were employed to incorporate shape cues that are inferred by SVM classification. While graph partitioning techniques are better equipped than active contour-based methods for the segmentation of sub-cellular structures, they have several practical issues. The first problem relates to the complexity of the undirected graphs in which closed contours are detected, and the second is the inference of graph searching constraints that accurately encode a set of rules that allows the identification of mitochondria membranes when dealing with the noisy and low contrast nature of the EM data.

Most recently, Ghita et al. [76] extended the framework proposed in this thesis to the grouping of superpixels. The authors investigated the segmentation of closed contours in subcellular data using a framework that also combined model-based spectral clustering (via the Affinity Factorization algorithm proposed by Perona and Freeman [166]) with a graph partitioning contour searching approach. One salient problem that precluded the application of spectral clustering methods to large scale segmentation problems is the onerous computational complexity required to generate comprehensive representations that include all pairwise relationships between all pixels in the input data. The authors suggest that in order to compensate for this problem, one has to reduce the complexity of the input data. This has been done by applying an oversegmentation technique prior to the application of the computationally demanding strands of the segmentation process. This approach opened the opportunity to build specific shape and intensity models that could be successfully employed to extract the salient membrane structures in the input image which are further processed to identify the cycles in an undirected graph. The authors applied the proposed framework to the segmentation of mitochondria membranes in electron microscopy data which are characterized by low contrast and low signal to noise ratio.

As can be seen from this review, most modern molecular segmentation tools at least partly use methods unified by the machine learning field. Some examples given above include classification with Support Vector Machines, Random Forest, AdaBoost and Gentle-Boost by training classifiers with manually annotated data. These approaches fall into the category of *supervised* machine learning tools. Some subcellular segmentation approaches use kernel-PCA, affinity factorization, graph-based spectral clustering and dimensionality reduction algorithms which belong to the class of *unsupervised* machine learning tools. Thus, below we review in detail the most popular machine learning algorithms in biomedical image processing and their theoretic foundations.

2.2 Machine Learning for Biomedical Imaging

Machine Learning techniques automatically learn from a set of examples how to classify new instances of the same type of data. In other words, the goal of machine learning is to make predictions on data inputs. The capacity to generalize, i.e. the ability to successfully classify unknown data and possibly infer generic rules or functions, is an important property of these approaches and is sought to be maximized. Machine learning tools fall mainly into two categories:

Unsupervised machine learning: The underlying structure of the training data, i.e. the desired output, is unknown and is to be determined by the training algorithm. For example, for a classification method this means that the class information is not available and has to be approximated by grouping the training examples using some distance measure, a technique called clustering.

Supervised machine learning: A training set and the corresponding desired outputs of the function to learn are available. Thus, during training the algorithm iteratively presents examples to the system and adapts its parameters according to the distance between the produced and the desired outputs.

2.2.1 Supervised Machine Learning

In the literature, many supervised learning algorithms have been applied to the segmentation tasks such as, for example, [19, 211, 63] and including [4]. Alomari et al. [4] used supervised learning with a back-propagation neural network and performed segmentation by classification of histopathology images. Feature reduction was done with PCA. However, their supervised learner required also training data, which is a tedious and time-consuming step.

On the other hand, unsupervised learning has been also applied to many segmentation problems. For example, Hall et al. [88] performed a comparison between a neural network and fuzzy clustering (unsupervised) techniques in segmenting magnetic resonance images of the brain. Liu et al. [128] improved the spacial spectral clustering technique to a non-local one and used it for image segmentation. They used the kernel k-means algorithm incorporated with the non-local spatial constraints and applied spectral clustering to their non-local spatial matrix for image segmentation. The authors then used synthetic and real images for testing and showed a high performance and reduced noise in their results.

In the application to digital pathology, Hiary et al. [92] proposed a set of clinically motivated features representing color, intensity, texture and location to segment and localize the tissue from the whole slide images using purely unsupervised learning with k-means clustering. The authors reported 96% localization accuracy on a large dataset. Furthermore, they concluded that because their method utilized unsupervised learning which does not require training, there was no need to fine-tune the parameters for each lab setting. Moreover, their method produced highly robust results comparable to supervised learning.

In regard to the automated subcellular segmentation, Narasimha et al. [153] proposed a machine learning tool for automatic texton-based joint classification and segmentation of mitochondria in microscopic images. Their approach is based on block-wise classification of images into a trained list of regions. Given manually labeled images, the goal is to learn models that can localize novel instances of the regions in test datasets. Classification was performed by k-Nearest Neighbor classifier, Support Vector Machines, adaptive boosting with AdaBoost and histogram matching using a Nearest Neighbor classifier. As with all texture-based machine learning algorithms, this method requires a considerable number of images for testing and training.

A different texture-based approach was reported by Vitaladevuni et al. [216]. Here authors considered the detection of mitochondria in TEM images of brain tissue and evaluated their method on the rat *neurophil* cell type. In particular, the authors applied the Gentle-Boost classifier which computes a mitochondria confidence map for each image plane. On the other hand, Nguyen et al. [156] proposed an energy-driven watershed method for the automatic segmentation of bacterial walls and mitochondrial boundaries.

Most relevant work to the approach developed in this work is outlined in [103] and combines spectral clustering (Normalized Cuts algorithm as in [190]), perceptual grouping and supervised machine learning to provide an automated segmentation solution to the extraction of membranes in neuroanatomy setting. In particular, the probability output of a Random Forest classifier is used in a regular cost function, which enforces gap completion via perceptual grouping constraints.

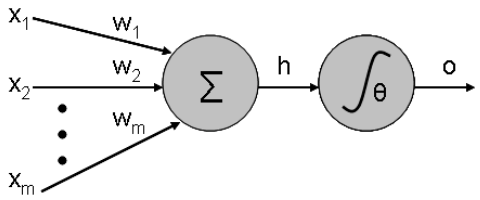


Figure 2.2.1: The Perceptron as a mathematical model of a neuron by McCulloch and Pitt [144]. The inputs x_i (dendrites) are multiplied by the weights w_i, and the neurons sum their values. If this sum is greater than the threshold θ then the neuron fires along the axons output; otherwise it does not. Image source: adapted from [144]

2.2.2 Neural Networks

Artificial neural networks, (or neural networks for short), denote a machine learning technique that has been inspired by the human brain and its capacity to perform complex tasks by means of inter-connected neurons performing each a very simple operation. Likewise, an artificial neural network is a trainable structure consisting of a set of inter-connected units, each implementing a very simple function, and together eventually performing a complex classification function or approximation task.

2.2.2.1 Perceptron

In biological terms, a *neuron* is a cell that can transmit and process chemical or electrical signals. The neuron is connected to other neurons to create a network. In graph theory this is equivalent to a graph with nodes and edges. Within humans, there are a huge number of neurons interconnected with each other - tens of billions of interconnected structures. Every neuron has an input (called a *dendrite*), a cell body, and an output (called the *axon*). Outputs connect to inputs of other neurons and the network grows. Biologically, neurons can have $10,000$ different inputs, but their complexity is much greater than their artificial analogues. Neurons are activated when the electrochemical signal is sent through the axon. The cell body determines the weight of the signal, and, if a threshold is passed, the firing of a pulse of fixed strength and duration continues through as output along the axon [18]. The axons divide (arborise) into connections to many other neurons, connecting to each of these neurons in a *synapse* [140]. Each neuron is connected to thousands of other neurons, so that it is estimated that there are about 100 trillion (= 10^{14}) synapses within the brain.

The best known mathematical model of a biological neural unit described above is called the Perceptron and was introduced by Rosenblatt [173]. Its basic structure is illustrated in Fig. 2.2.1, which depicts the McCulloch and Pitts [144] neuronal model [140] which contains:

1. **a set of weighted inputs** w_i that correspond to the synapses
2. **an adder** that sums the input signals (equivalent to the membrane of the cell that collects electrical charge)
3. **an activation function** (initially a threshold function) that decides whether the neuron fires ('spikes') for the current inputs

Generally this model has m inputs and one output where the output is a simple function of the sum of the input signals $\mathbf{x}$ weighted by $\mathbf{w}$ and an additional bias b:

$$y = g(\mathbf{x} \cdot \mathbf{w} + b) = g(h + b), \tag{2.1}$$

where

$$h = \sum_{i=1}^{m} w_i x_i, \tag{2.2}$$

Often, the bias b is put inside the weight vector $\mathbf{w}$ such that $w_0 = b$ and the input vector $\mathbf{x}$ is extended correspondingly to have $x_0 = 1$. Equation 2.1 then becomes:

$$y = g(\mathbf{x} \cdot \mathbf{w}) \tag{2.3}$$

where the activation function g is given as the Heavyside step function:

$$g : \mathbb{R} \to \mathbb{R}, \qquad g(x) = \begin{cases} 1, & \text{if } x \geq 0; \\ 0, & \text{otherwise.} \end{cases} \tag{2.4}$$

The Perceptron thus implements a very simple two-class classifier where $\mathbf{w}$ is the separating hyperplane such that $\mathbf{w} \cdot \mathbf{x} >= 0$ for examples from one class and $\mathbf{w} \cdot \mathbf{x} < 0$ for examples from the other. In the neuronal model as in Fig.2.2.1 the weights were labeled as w_i, with the i index running over the number of inputs. Here, we also need to work out which neuron the weight feeds into, so we label them as w_{ij}, where the j index runs over the number of neurons.

The Perceptron Learning Algorithm has two major parts: a *training* phase and a *recall* phase. The training phase uses the recall equation (2.7), since it has to work out the activations of the neurons before the error can be calculated and the weights trained [140]. Given that y_j is the output and t_j is a target neuron, the simple error function can be written as $(y_j - t_j)$. The learning rate parameter η in (2.6) decides how fast the network learns. The Perceptron Learning Algorithm is summarized in Table 2.2.1.

Table 2.2.1: Perceptron Learning Algorithm

1	Set all the weights w_{ij} to small (positive and negative) random numbers
2	For T iterations or until all the outputs are correct and for each input vector x_i, compute the activation of each neuron j using activation function g using (2.5)
3	Update each of the weights individually using (2.6)
4	Compute the activation of each neuron j using (2.7)

$$y_j = g(\sum_{i=0}^{m} w_{ij} x_i) = \begin{cases} 1, & \text{if } \sum_{i=0}^{m} w_{ij} x_i > 0 \\ 0, & \text{if } \sum_{i=0}^{m} w_{ij} x_i \leq 0 \end{cases} \tag{2.5}$$

$$w_{ij} \leftarrow w_{ij} - \eta(y_j - t_j) \cdot x_i \tag{2.6}$$

$$y_j = g(\sum_{i=0}^{m} w_{ij} x_i) = \begin{cases} 1, & \text{if } w_{ij} x_i > 0 \\ 0, & \text{if } w_{ij} x_i \leq 0 \end{cases} \tag{2.7}$$

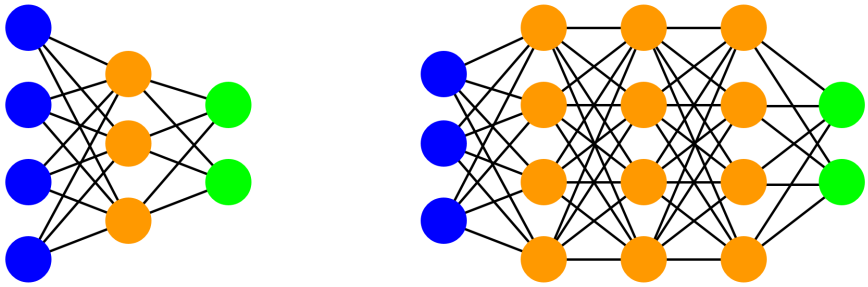


Figure 2.2.2: Example of two different Multi Layer Perceptron architectures. Both shown MLPs are fully connected feed-forward networks. The left architecture has the topology (4-3-2). Input nodes are shown in blue, the three nodes of the single hidden layer are shown in orange and the nodes in the output layer are shown in green. An example of a deep MLP is shown in the right image. This network has three hidden layers and has the topology (3-4-4-4-2). The MLP is also the key element of a CNN (Convolutional Neural Network). Image sources: adapted from [214]. Diagrams are best viewed in color.

In 1962, Rosenblatt introduced the perceptron convergence theorem [174], a supervised training algorithm capable of learning arbitrary two-class classification problems. However, Minsky and Papert [145] pointed out that there are very simple classification problems where the perceptron fails. This is a case in the XOR-problem, where the pattern (0, 0) and (1, 1) belong to one class and (0, 1) and (1, 0) to the other. These two classes are not linearly separable. This fact has motivated the use of several interconnected perceptrons which are able to form more complex decision boundaries by combining several hyperplanes. The most common type of such neural networks is the Multi Layer Perceptron.

2.2.2.2 Multi Layer Perceptron

A single biological neuron and its mathematical Perceptron model are very simple structures. Only a *network* of neurons in the brain can perform complicated operations. Therefore, the structure of a neural network is a very crucial matter. Multi Layer Perceptrons are capable of approximating arbitrarily complex decision functions. With the advent of a practicable training algorithm in the 1980's, the so-called Backpropagation algorithm [177], they became the most widely used form of neural networks. Fig. 2.2.3 provides an example of two distinct architectures of a MLP.

In the MLP architecture there is an input layer, one or more hidden layer(s) and an output layer of neurons, where each neuron except the input neurons implements a perceptron as described in the previous section. Moreover, the neurons of one layer are only connected to the following layer. This type of network is called feed-forward network, where the activation of the neurons is propagated layer-wise from the input to the output layer. If there is a connection from each neuron to every neuron in the following layer, as in Fig. 2.2.3, the network is called *fully-connected*. Further, the neurons' activation function has to be differentiable in order to adjust the weights by the Backpropagation algorithm. Commonly used activation functions are for example:

$$g(x) = x, \text{ linear} \tag{2.8}$$

$$g(x) = \frac{1}{1 + e^{-cx}}, (c > 0), \text{ sigmoid} \tag{2.9}$$

$$g(x) = \frac{1 - e^{-x}}{1 + e^{-x}}, \text{ hyperbolic tangent} \tag{2.10}$$

The MLP is one of the most common neural networks in use. This is due to the fact that the Perceptron algorithm can only solve linearly-separable problems. The majority of real-world problems, however, are non-linear [140]. The MLP learning algorithm consists of two parts. The first part calculates the outputs for the given inputs and the current weights. The objective of learning is to minimize the error and the learning in the neural network happens in the weights. Therefore, the second part updates the weights according to the error, which is a function of the difference between the outputs and the targets. This process is called the back-propagation of error through the network. This means that errors are sent backwards through the neural network.

In the Perceptron model the weights are changed in such a way that the neurons fired when the targets said they should and did not fire when the targets said they should not. For that purpose we define an error function for each neuron $k : E_k = y_k - t_k$, and try to make it as small as possible. In order to quantify the error we can use the sum-of-squares error function such as:

$$E(\mathbf{t}, \mathbf{y}) = \frac{1}{2} \sum_{k=1}^{N} (y_k - t_k)^2 \tag{2.11}$$

The gradient of $E(\mathbf{t}, \mathbf{y})$ is obtained by differentiation. The gradient descent concept implies that the error goes downhill until it reaches a local minimum. The weights of the network are trained in order to achieve the minimization of the error. The complete mathematical description of the MLP algorithm is provided in [140] and consists of initialization, training (forwards and backwards phases) and recall components. Table 2.2.2 includes a short summary of the major operations.

Table 2.2.2: Multi Layer Perceptron Learning Algorithm

1	An input vector is put into the input nodes
2	The inputs are fed forward through the network
	2.1 The inputs and the first-layer weights are used to decide whether the hidden nodes fire or not. One can use the sigmoid function (2.9) which is differentiable
	2.2 The outputs of these neurons and the second layer weights are used to decide if the output neurons fire or not
3	The error is computed according to (2.11) as the sum-of-squares difference between the network outputs y and the targets t
4	This error is fed backwards through the network in order to:
	4.1 First update the second layer weights
	4.2 Afterwards update the first layer weights

In order to use the MLP to solve real world problems, a few issues have to addressed. First, Marsland in [140] explains the criteria needed to terminate the training process. The technique is called *early stopping*. Generally, the training process should continue until a local minimum has been found. At the same time, keeping on training too long leads to overfitting of the network. In order to overcome this issue it is advised to use the validation set to estimate how well the network is generalizing. The recipe is to carry on training for a few iterations and then repeat the whole process. At some stage the error on the validation set starts increasing again, because the network has stopped learning about the function that generated the data, and started to learn about the noise that is in the data itself. This is the stage where to stop training.

Secondly, how much training data is needed depends on the problem. One rule is to use a number of training examples that is at least 10 times the number of weights. This can be a very large number of examples leading to the neural network training being a computationally expensive operation because it is necessary to show the network all of these inputs lots of times. Thirdly, the number of hidden nodes and the number of hidden layers is important. Marsland in [140] explained that two hidden layers is the most needed for normal MLP learning as it is possible to show mathematically that one hidden layer with lots of hidden nodes is sufficient. As to the choice of the number of hidden nodes, there is no clear advice. Here it is possible to train several networks with different number of hidden nodes and choose the one that gives the best results.

Neural networks were extensively studied in the 1980s and early 1990s, but with mixed empirical success. In recent years, a combination of algorithmic advancements, as well as increasing power and data size, has led to a breakthrough in the effectiveness of neural networks. In particular, deep NNs (i.e., networks of more than two layers) have shown very impressive practical performance on a variety of domains as has been noted by Shalev-Shwartz and Ben-David [186].

2.2.2.3 Convolutional Neural Networks and Deep Learning

The origin of the applications of deep learning to object recognition and image segmentation tasks can be traced to the Convolutional Neural Networks in the early 1990s. However, the success of the early CNNs was limited due to the size of the available training sets and the size of the networks considered [172]. CNN-based architectures have captured intense interest in computer vision since October 2012 after the ImageNet competition results were released. The winning CNN architecture proposed by Krizhevsky et al. [112] consists of a large network with 8 layers, $650,000$ neurons and 60 millions of parameters trained on the ImageNet dataset with 1.2 million high-resolution training images. The authors reported a huge recognition accuracy gain over competing approaches by using GPU-like high-performance computing platforms [46]. Their network takes between five and six days to train on two GTX 580 3GB GPUs. Since then, even larger and deeper networks have been developed and trained [172].

Leena and Govindan [121] presented a novel CNN-based segmentation of EM data. The CNN developed is able to extract features directly from pixel images with minimal preprocessing. The authors claim that it can even recognize a pattern which has not been presented before, provided it resembles one of the training patterns. After learning from ground-truth images, the CNN automatically generates an affinity graph from raw microscopy images. This affinity graph can be then paired with any standard partitioning algorithm, such as Ncut [190] or the Connected Component algorithm, to achieve segmentation. The F-score of this approach is reported to be 78%.

Generally, CNNs consist of a set of layers, and each layer contains one or more planes. The input to each unit in a plane is accepted from a small neighborhood in the planes of the previous layer. The *shared weight* concept is applicable to each plane, and multiple planes in each layer detect multiple features. After detecting the features, the image is passed to a subsampling layer which is used to perform a local averaging of the weights. Shared weights help to reduce the number of parameters of the network [121]. The first step is to generate an affinity graph. Most affinity functions used to design the edge weights depend on local image features such as intensity,

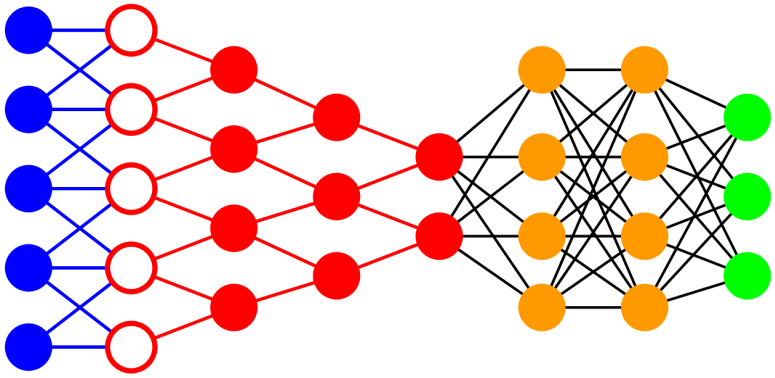


Figure 2.2.3: Example of a deep CNN architecture which comprises eight layers in total. The input layer is followed by a kernel layer (red circle). Three convolution (or pool) layers are shown in red. It can be seen that the first five layers are locally connected. The last three layers are fully connected. Image source: adapted from [214]. This diagram is best viewed in color.

spatial derivatives, texture or color. The CNN is then trained on the raw input image and the ground truth segmentation generated by human experts. The authors observe that supervised graph methods outperform their unsupervised peers in various classification tasks. Supervised graphs are typically constructed by allowing two nodes to be adjacent only if they are of the same class. However, the performance of a graph-based algorithm can be diminished by poor choices of affinity function. Nowlan and Platt [160] have introduced a novel CNN to track a hand in a sequence of video frames. This gesture recognition system is able to classify whether the hand is closed or open in 99.1 % of the test frames. This system can detect a hand in 99.7 % of the test frames. Garcia and Delakis [74] have successfully used Convolutional Neural Networks on face detection and reported a perfect recognition rate. The system developed can detect highly variable face patterns, even with rotated images. Lawrence et al. [119] proposed a novel CNN topology by using a 1D Haar wavelet in the first layer. The convolutional face features for the MLP classifier is then used to recognize the face. In another work, the translation invariance feature of CNN was used by Waibel et al. [218] to recognize the dynamic structure of phonemes. LeCun et al. [120] showed that the CNN outperforms all other techniques as it can recognize the variability in characters, and they proposed a perfect hand-written character recognizer based on CNNs. There are many more CNN architectures proposed for various applications. The dissertation by Saidane [178] gives a good review of CNN architectures for the years before 2011. A recent review is given by Ronnenberber et al. [172] who won the ISBI cell tracking challenge in 2015. The authors developed a modified CNN and a new training strategy for biomedical image segmentation so that it can work with very few training images. They showed impressive results on the segmentation of neuronal structures in EM stacks as well as cell segmentation in light microscopy images. With very few annotated images such a CNN has a very "reasonable" training time of only 10 hours on a NVidia Titan GPU (6 GB) platform.

2.2.2.4 Neocognitron

In the neocognitron model proposed by Fukushima in [72], a network acquires a structure similar to the hierarchy model of the mammal's visual nervous system proposed by Hubel and Wiesel [95].

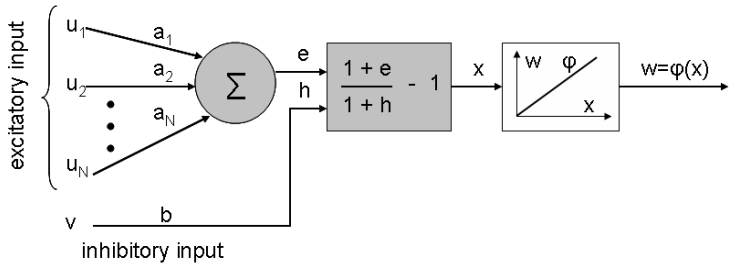


Figure 2.2.4: Input-output characteristics of an S-cell: an example of the cells used in the neocognitron model which has been developed by Fukushima in [73]. Image source: adapted from [73]

According to the hierarchy model of the cat's visual cortex, the neural network in the visual cortex has the following hierarchy structure :

LGB (lateral geniculate body) → simple cells → complex cells → lower order hypercomplex cells → higher order hypercomplex cells.

The neural network between lower order hypercomplex cells and higher order hypercomplex cells has a structure similar to the network between simple cells and complex cells. A cell in a higher stage has a tendency to respond selectively to a more complicated feature of the stimulus pattern.

The neocognitron-based network, as proposed by Fukushima in [72], consists therefore of layers of *simple* cells corresponding to feature extracting cells called S-cells, and layers of *complex* cells corresponding to recognition cells called C-cells. The layout of an S-cell can be seen in Fig. 2.2.4. These layers of S- and C-cells are arranged alternately in a hierarchical manner into a network. Furthermore, S-cells or C-cells in a layer are sorted into subgroups according to the optimum stimulus features of their receptive fields. Fukushima uses the terminology of S-plane and C-plane representing cell-planes consisting of S-cells and C-cells, respectively. The network in [72] has forward and backward connections between cells as explained in [178]. In this hierarchy, the forward signals manage the function of pattern recognition, while the backward signals manage the function of selective attention, pattern segmentation, and associative recall. Some of the connections between cells are variable, and the network can acquire the ability to recognize patterns by unsupervised learning. A model of an S-cell is provided in Fig. 2.2.4. Besides the input from the receptive field, called the excitatory input e, there is also an inhibitory input h which has a negative effect on the activation of the neuron [60]. If h is greater than e, the output of the neuron is zero. The excitatory input is calculated as follows:

$$e = \sum_{i=1}^{N} a(i)u(i) \tag{2.12}$$

where $a(i)$ are the training weights, $u(i)$ denote the inputs from the preceding cells, and N is the number of weights. The S-cell receives variable excitatory connections from a group of C-cells of the preceding layer. The cell also receives a *variable* inhibitory connection from an inhibitory cell, called a V-cell. The V-cell receives *fixed* excitatory connections from the same group of C-cells as does the S-cell, and always responds with the average intensity of the output of the C-cells.

The inhibitory input h is calculated as follows:

$$h = bv \tag{2.13}$$

where b is a trainable weight. In the neocognitron, the input v is calculated as the weighted root-mean-squared values coming from the receptive field and thus represents some kind of normalization [60]. The activation of a S-cell is:

$$u_S(i) = \phi\left(\frac{1+e}{1+h} - 1\right) \tag{2.14}$$

where the activation function ϕ is defined as:

$$\phi(x) = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.15}$$

Fukushima [73] showed that the outputs of the S-cells approximate a *convolution* normalized by the length of the weight vector and the input vector. Several S-planes, each containing a different set of weights, can be used to extract different features at the same locations. The set of S-planes at one particular level form a S-layer. The exact position of each feature is not very important in most cases. Therefore, each S-plane is followed by a C-plane which reduces the resolution of the respective S-plane by a constant factor, e.g. two and thus performs a kind of sub-sampling or blurring. The sub-sampling also reduces the sensitivity of the neural network to small shifts and distortions of the input pattern. By alternating C-layers and S-layers and combining the outputs of the respective maps, one can construct more complex feature extractors. In the neocognitron the first S-planes extract simple visual features, such as oriented edges or corners, and the following layers combine them to extract more complex features, such as combinations of line segments [60].

2.2.2.5 Architecture of the *LeNet-5* CNN

In the nineties, LeCun and his colleagues [120] developed a series of convolutional neural networks, called LeNet-5. Their network architecture, shown in Fig. 2.2.5, consists of a cascade of feature detection layers. Instead of having an S-layer followed by a C-layer, the networks have alternating convolution and sub-sampling layers. Moreover, the model of the individual neurons is the basic perceptron with the sigmoid activation function, which was not the case in the neocognitron. Similarly to the neocognitron, these CNNs are based on three architectural ideas [178]:

Table 2.2.3: Architectural ideas of the LeNet-5 CNN.

Local receptive fields:	Inspired by the mammalian visual cortex and which are used to detect elementary visual features in images
Shared weights:	Extract the same set of elementary features from the whole input image and reduce the computational cost
Sub-sampling:	Operations which reduce the computational cost and the sensitivity to affine transformations such as shifts and rotations

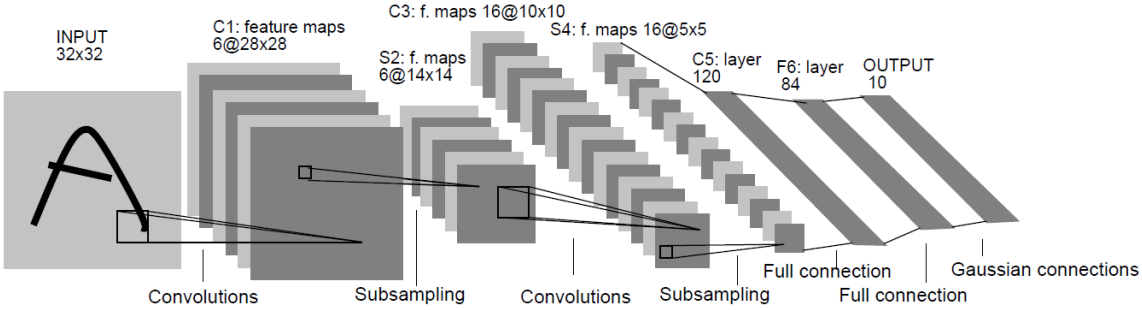


Figure 2.2.5: Architecture of LeNet-5, a Convolutional Neural Network designed by LeCun and his colleagues for digits recognition application. In this architecture, each plane is a feature map with a set of units whose weights are constrained to be identical. Image source: [120]

Table 2.2.4: Parameters of the LeNet-5 Convolutional Neural Network.

	C1	S2	C3	S4	C5
Feature maps	6	6	16	16	120
Size of feature maps	28 x 28	14 x 14		5 x 5	
Neighborhood	5×5	2×2	5×5	2×2	5×5
Total number of trainable parameters	156	12		32	
Total number of connections	122,304	5,880		2000	48,120

An important difference between the Fukushima and the LeCun models concerns the training procedure. LeCun and his team were the first to apply back-propagation to CNNs [178]. The architecture of LeNet-5 corresponds to a feed-forward MLP network and comprises seven layers, not counting the input, all of which contain trainable parameters. Each layer of the first five layers contains one or more planes where each plane is made up of a 2D lattice of neurons. Each neuron is locally connected to a set of neurons located within its local receptive field in the planes of the previous layer. These planes are called feature maps. We summarize the parameters of the LeNet-5 CNN in Table 2.2.4. All the neurons in feature maps of C1 share the same set of 25 tunable weights constituting a 5×5 trainable kernel and a bias. Each convolution map in C1 is followed by a subsampling map which performs averaging and reduces the dimension of the respective convolution map by a factor of two. The 2×2 receptive fields are non-overlapping, therefore feature maps in S2 have half the number of rows and columns of feature maps in C1. Layers C3 and S4 are implemented similarly to layers C1 and S2 respectively. The last hidden layer has 84 neurons and has full connection with C5. In total, the LeNet-5 architecture has 345,308 connections, but due to the weight-sharing mechanism, there are only 60,000 free parameters [120, 178].

The most notable advance using CNNs was achieved in the 2012 ImageNet LSVRC competition, in which the task was to train a model with 1.2 million high-resolution images to classify unseen images to one of the 1000 different image classes. On the test set consisting of 150k images, the deep CNN developed by Krizhevsky et al. [112] achieved error rates considerably lower than the previous state of the art. Very large deep CNNs were used, consisting of 60 million weights, 650,000 neurons, five convolutional layers together with max-pooling layers and two fully-connected layers on top of the CNN layers [46].

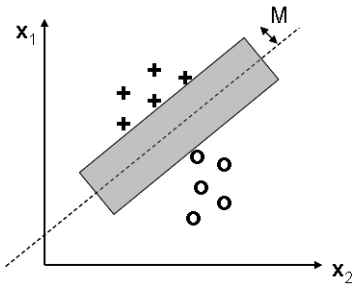


Figure 2.2.6: The idea of SVM classifiers. The margin M is the largest region that separates the two classes without there being data points inside. The separation box is made from two lines that are parallel to the decision boundary. Image source: adapted from [140].

2.2.3 Support Vector Machines

A Support Vector Machine (SVM) is essentially a technique for classifying objects. Support vector machines are used in a variety of classification scenarios, such as image recognition and handwriting pattern recognition. The SVM is according to Marsland [140] one of the most popular algorithms in modern machine learning. SVMs were introduced by Boser et al. in 1992 [23] and modified by Cortes and Vapnik in 1995 [39].

Image classification can be greatly improved with the use of support vector machines. Being able to classify thousands or millions of images is becoming more and more important with the use of smartphones and applications like Instagram as noted by Jason Bell [18]. Support vector machines can also do text classification on normal text or web documents, for instance. Machine learning with support vector machines takes the concept of a perceptron a little bit further to maximize the geometric margin. It is one of the reasons why SVM and artificial neural networks are frequently compared in function and performance.

The classifier in Fig. 2.2.6 has a large linear margin M. The datapoints in each class that lie closest to the classification line are called support vectors. The best classifier is the one that goes through the middle of space between two classes '+' and 'o'. Therefore the margin should be as large as possible and the support vectors are the most useful datapoints because they are the ones that we might get wrong. In order to proceed with a formal description of these ideas we first define a weight vector $\mathbf{w}$ (a vector, not a matrix, since there is only one output) and an input vector $\mathbf{x}$. The output y that we defined in the Perceptron section in the equation (2.1) is equal to $y = \mathbf{w} \cdot \mathbf{x} + b$ where the "$\cdot$" denotes the dot product (also called inner or scalar product) between $\mathbf{w}$ and $\mathbf{x}$, and b is the contribution from the bias weight as defined in the Perceptron model and a threshold in the Support Vector Machine model. In this section we use the notation for the dot product - $\langle \mathbf{w}, \mathbf{x} \rangle$ as used by Schölkopf [182].

Suppose we are given a dot product space $\mathcal{H}$, and a set of pattern vectors $\mathbf{x}_1, ..., \mathbf{x}_m \in \mathcal{H}$. We then define a class of hyperplanes in $\mathcal{H}$ by:

$$\{\mathbf{x} \in \mathcal{H} \mid \langle \mathbf{w}, \mathbf{x} \rangle + b = 0\} \tag{2.16}$$

where $\mathbf{w} \in \mathcal{H}, b \in \mathbb{R}$, corresponding to decision functions

$$f(x) = \text{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle + b) \tag{2.17}$$

In this formulation, $\mathbf{w}$ is a vector orthogonal to the hyperplane: if $\mathbf{w}$ has unit length, then $\langle \mathbf{w}, \mathbf{x} \rangle$ is the length of $\mathbf{x}$ along the direction of $\mathbf{w}$. For general $\mathbf{w}$, this number is scaled by $\|\mathbf{w}\|$. In any case, the set (2.16) consists of vectors that all have the same length along $\mathbf{w}$. In other words, these are vectors that project onto the same point on the line spanned by $\mathbf{w}$. Among all hyperplanes separating the data, there exists a unique optimal hyperplane with the maximum margin of separation between any training point and the hyperplane. It is the solution of the optimization problem:

$$\max_{\mathbf{w}\in\mathcal{H}, b\in\mathbb{R}} \min\{\|\mathbf{x} - \mathbf{x}_i\| \mid \mathbf{x} \in \mathcal{H}, \langle \mathbf{w}, \mathbf{x} \rangle + b = 0, i = 1, ..., m\} \tag{2.18}$$

An example illustrated in Fig. 2.2.7 shows how the linear classification works. Any $\mathbf{x}$ value that gives a positive value for $\langle \mathbf{w}, \mathbf{x} \rangle + b$ is above the line, and so is an example of the "+" class, while any $\mathbf{x}$ that gives a negative value is in the "o" class. In order to include the margin M into computation, instead of just looking at whether the value of $\langle \mathbf{w}, \mathbf{x} \rangle + b$ is positive or negative, we also check whether the absolute value is less than the margin M, which would put it inside the grey box in Fig. 2.2.6. According to Schölkopf [182], for a given hyperplane $\{\mathbf{x} \in \mathcal{H} | \langle \mathbf{w}, \mathbf{x} \rangle + b = 0\}$, we call:

$$\mathrm{M}_{(\mathbf{w},b)}(\mathbf{x}, t) := t(\langle \mathbf{w}, \mathbf{x} \rangle + b)/\|\mathbf{w}\| \tag{2.19}$$

the geometrical margin of the point $(\mathbf{x}, t) \in \mathcal{H} \times \{\pm 1\}$. For a given margin value M (see Fig. 2.2.6):

- any point $\mathbf{x}$ where $\langle \mathbf{w}, \mathbf{x} \rangle + b \geq \mathrm{M}$ belongs to the "+" class,
- any point $\mathbf{x}$ where $\langle \mathbf{w}, \mathbf{x} \rangle + b \leq -\mathrm{M}$ belongs to the "o" class,
- the separating hyperplane is specified by $\langle \mathbf{w}, \mathbf{x} \rangle + b = 0$ (see also Fig. 2.2.7),
- a support vector is for example a point $\mathbf{x}^+$ that lies on the "+" class boundary line, so that $\langle \mathbf{w}, \mathbf{x}^+ \rangle = \mathrm{M}$.

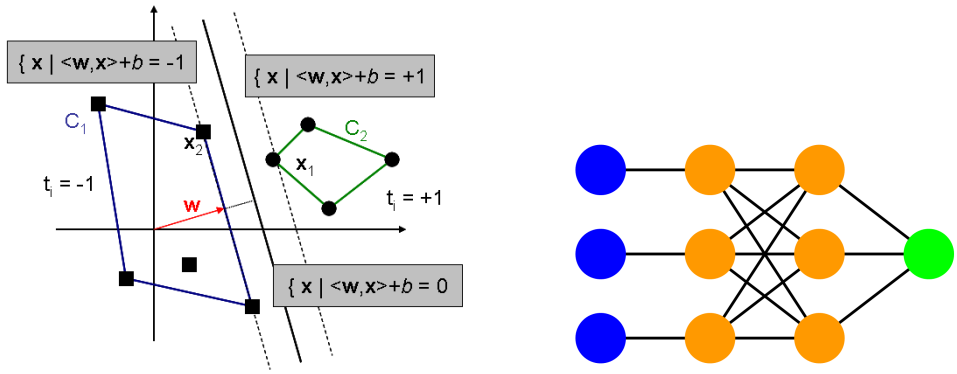
The pair $(\mathbf{w}, b) \in \mathcal{H} \times \mathbb{R}$ is called a *canonical* form of the hyperplane (2.16) with respect to $\mathbf{x}_1, ..., \mathbf{x}_m \in \mathcal{H}$ if it scaled such that:

$$\max_{i=1,...,m} |\langle \mathbf{w}, \mathbf{x}_i \rangle + b| = 1 \tag{2.20}$$

which amounts to saying that the point closest to the hyperplane has a distance of $1/\|\mathbf{w}\|$. Note that the condition (2.20) allows two such pairs: given a canonical hyperplane $(\mathbf{w}, b)$, another one satisfying (2.20) is given by $(-\mathbf{w}, -b)$. For the purpose of pattern recognition, these two hyperplanes are different as they are oriented differently and thus they correspond to two decision functions:

$$f_{\mathbf{w},b} : \mathcal{H} \to \{\pm 1\} \qquad \mathbf{x} \mapsto f_{\mathbf{w},b}(\mathbf{x}) = \mathrm{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle + b) \tag{2.21}$$

which are inverse of each other. In the absence of class labels $y_i \in \{\pm 1\}$ associated with the $\mathbf{x}_i$, there is no way of distinguishing the two hyperplanes. For a labeled dataset, a distinction exists: the two hyperplanes make opposite class assignments. In pattern recognition, we attempt to find a solution $f_{\mathbf{w},b}$ which correctly classifies the labeled examples $(\mathbf{x}_i, y_i) \in \mathcal{H} \times \{\pm 1\}$; in other words, which satisfies $f_{\mathbf{w},b}(\mathbf{x}_i) = y_i$ for all i. In this case the training set is said to be separable [182].



(a) (b)

Figure 2.2.7: (a) The optimal hyperplane is the one bisecting the shortest connection between the convex hulls C_1 and C_2 of the two classes. (b) An example of a neural network which can perform kernel-SVM by non-linearly mapping input vectors to a very high-dimensional space where the linear decision boundary can be constructed. Image sources: adapted from [182] and [214].

2.2.3.1 Kernel-SVM

The SVM algorithm computes the optimal margin hyperplane and the support vectors $\mathbf{x}_i$ which lie directly on the margin. The related optimization problem and its solutions are provided in [140, 146, 186]. However, the SVM algorithm has a major drawback in that it assumes the linear separability of the underlying dataset. To allow for much more general decision surfaces, we can use kernels to nonlinearly transform the input data $x_1, ..., x_n \in \mathcal{X}$ into a high-dimensional feature space using a map $\Phi : x_i \mapsto \phi(\mathbf{x}_i)$ and then perform a linear separation there. According to Schölkopf [182], such a kernel mapping increases the chances of a separation because using a kernel means using a larger function class. Therefore, this increases the capacity of the learning machine and finds a linear decision boundary that separates the classes in the feature space.

We first introduce new functions $\phi(\mathbf{x})$ of some input features. Then, in order to use the SVM algorithm for the prediction of the new test point $\mathbf{z}$, we replace $\mathbf{x}_i$ by $\phi(\mathbf{x}_i)$ and $\mathbf{z}$ by $\phi(\mathbf{z})$ such that:

$$\mathbf{w}^T\mathbf{z} + b = \left(\sum_{i=1}^{n} \lambda_i t_i \phi(\mathbf{x}_i)\right)^T \phi(\mathbf{z}) + b \tag{2.22}$$

where λ_i are Langrange multipliers, $\mathbf{x}_i$ are the support vectors for which $\lambda_i > 0$ and t_i are the known targets. Further, given a function $k : \mathcal{X}^2 \to \mathbb{K}$ and $x_1, ..., x_n \in \mathcal{X}$, the $n \times n$ matrix $\mathbf{K}$ with elements:

$$K_{ij} := k(x_i, x_j) \tag{2.23}$$

is called the kernel (or Gram) matrix of k with respect to $x_1, ..., x_n$. Thus, $\mathbf{K}$ is made from the dot product of the original vectors. The evaluation of the decision function $f(\mathbf{x})$ requires computation of dot products $\langle \Phi(x), \Phi(x_i) \rangle$ in a high-dimensional space using a positive definite kernel k:

$$\langle \Phi(x), \Phi(x_i) \rangle = k(x, x_i) \tag{2.24}$$

With $k(x, x_i)$ we obtain decision functions of the form:

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^{n} t_i \lambda_i k(x, x_i) + b\right) \tag{2.25}$$

Any symmetric function that is positive definite can be used as a kernel. In the literature, such kernels are frequently called PDS (Positive Definite Symmetric) kernels [146]. Note that a symmetric matrix is positive definite if and only if all its eigenvalues are nonnegative. It is also possible to convolve kernels together to obtain another kernel. The most widely used kernels are:

- Polynomial classifiers of degree γ:

$$k(x, x_i) = \langle x, x_i \rangle^{\gamma} \qquad \text{or} \qquad k(x, x_i) = (1 + \langle x, x_i \rangle)^{\gamma} \tag{2.26}$$

  with $\gamma = 1$ for a linear kernel

- Radial Basis Function classifier with Gaussian kernel of width $c > 0$ ($c = 2\sigma^2$):

$$k(x, x_i) = \exp(-\|x - x_i\|^2 / c) \tag{2.27}$$

- Neural Networks with tanh activation function:

$$k(x, x_i) = \tanh(\kappa \langle x, x_i \rangle + \theta) \tag{2.28}$$

where the parameters $\kappa > 0$ and $\theta \in \mathbb{R}$ are the gain and horizontal shift respectively [182]. This kernel is also called the *sigmoid* kernel. Using sigmoid kernels with SVMs leads to an algorithm that is closely related to learning algorithms based on simple neural networks which are also often defined via a sigmoid function [146]. In most application cases, the best kernel and its parameters are found by experimenting with different values using a validation set.

The commonly used polynomial or Gaussian kernels are all PDS kernels over vector spaces. In many learning tasks found in practice, the input space $\mathcal{X}$ is not a vector space. The examples to classify in practice could be protein sequences, images, graphs, parse trees, finite automata, or other discrete structures which may not be directly given as vectors. For these types of problems the class of *sequence* kernels have been developed and applied [146]. Another domain of problems includes multi-class classification, such as OCR (Optical Character Recognition) problem where the class labels can take more than ± 1 binary values. Modified SVMs have been successfully applied in other computer vision tasks, which relate to the OCR problems. Examples include object and face detection and recognition and image retrieval [182].

2.2.4 AdaBoost

The AdaBoost algorithm is conceptually very simple. It is an algorithm that sequentially selects weak classifiers from a candidate pool and weights each of them based on their error. Each iteration of AdaBoost assigns an importance weight to each example. Examples with a higher weight, classified incorrectly on previous iterations, will receive more attention on subsequent iterations, tuning the weak learners to difficult examples. Testing examples with AdaBoost is therefore simply a weighted vote of the weak learners. Therefore, AdaBoost can be seen as a general method for improving accuracy of any given learning algorithm. Also, AdaBoost can produce very complex non-linear decision boundaries (closed contour) by combining several weak classifiers [140].

In the context of biomedical imaging, AdaBoost has been used in [193, 153] for cells, in [193, 153, 125] for mitochondria and in [14, 154] for synapses segmentation. There is a reason why these authors preferred using AdaBoost rather than SVMs. As we have seen above, SVMs seek a hypersurface in the space of all features that both minimizes the error of training examples and maximizes the margin, defined as the distance between the hypersurface and the closest value in feature space. Kernel-SVMs can use any type of hypersurfaces by making use of the kernel trick. SVMs are also quicker to train and evaluate than AdaBoost [148]. On the other hand, AdaBoost can select informative features from a potentially very large feature pool. This is based on the fact that AdaBoost belongs to the class of ensemble learning algorithms which make use of several weak classifiers. A weak learner is any statistical classifier that does not perform perfectly when used on its own, but still performs better than pure chance. The idea is that by putting the weak learners together it is possible to make an ensemble learner that can perform arbitrarily well.

Cascaded AdaBoost classifiers have been applied, for example by Smith et al. [193] for mitochondria segmentation. The authors praised the efficiency of AdaBoost in detecting objects in images. Smith and his team used their proposed Ray features to define a weak learner and to train a boosted classifier. Furthermore, AdaBoost was used to reject some negative samples while passing on positive samples to the next more discriminative stage.

Gentle-Boost is one of the many extensions to AdaBoost developed. Gentle-Boost allows us to increase performance of a classifier and to reduce computation in comparison to AdaBoost. Vitaladevuni et al. [216], for example, used Gentle-Boost in their approach to detecting mitochondria in EM images. The authors concluded that Gentle-Boost was well suited for handling large numbers of potentially irrelevant and redundant features. The classifier computed a mitochondria confidence map for each image plane. Pixels with high values were likely to belong to mitochondria.

2.2.5 Random Forest

Random Forest is another popular supervised classification technique which is frequently used for biomedical image segmentation [103, 111, 184, 203, 229]. Also, the Random Forest classifier is frequently used with superpixels [111, 203, 229] which reduce the complexity of the input data.

Random Forest, similar to AdaBoost, belongs to the class of ensemble learners. The core idea is that if one tree is good, then many trees should be better, provided that there is enough variety between them. The most prominent fact about the Random Forest is how it creates randomness from a standard dataset. In order to create a forest, the trees can be made different by training them on slightly different data. An additional way to add more randomness is to limit the choices that the decision tree can make. At each node, a random subset of the features is given to the tree and it can only pick from that subset rather than from the whole set. As well as increasing the randomness in the training of each tree, this approach also speeds up the training [140].

There is a conceptual algorithmic difference between AdaBoost and Random Forest. Classification with AdaBoost is expensive because this algorithm searches over the whole set of features at each stage, where each stage depends on the previous one. Therefore, boosting has to run sequentially, and the individual steps can be expensive. In contrast, the parallelism of the Random Forest and the fact that it only searches over a fairly small set of features at each stage, speed the algorithm

up a lot. This is the reason that Random Forest can deal very well with very large datasets.

For the cell detection and segmentation tasks, Zhang et al. [229] firstly computed a cell boundary probability map from a trained edge Random Forest classifier. Furthermore, they constructed superpixels and built a weighted superpixel adjacency graph. Segmentation is then generated from partitioning this graph using their proposed correlation clustering procedure which belongs to a spectral clustering class of algorithms. Seyedhosseini et al. [184] developed a mitochondria segmentation approach which started with the extraction of overlapping patches with different sizes for a given image. Next, polynomials of different degrees were fitted to each patch and the shape and textural features were computed for each patch. These features were then passed to the Random Forest classifier. If a patch was classified as positive by the Random Forest classifier, all the connected pixels of the center pixel in that patch were marked as mitochondria in the input image.

In their work on cell segmentation in EM datasets, Tek et al. [203] used the open source tool *ilastik* to interactively train a Random Forest classifier. Then the authors performed thresholding and connected component analysis on the classifier output probabilities. The authors noted that further improvement in classification performance could be achieved by using more training data for the final stage classifier. Another avenue for improvement could be to enhance nuclear staining through the use of nuclear specific heavy metal stains. Kaynig and her colleagues [103] used the probabilistic output of the Random forest classifier trained on annotated data for membrane detection. The authors incorporated this output into an energy cost function that was further minimized in the Graph Cut framework.

2.2.6 Unsupervised Machine Learning

The goal of supervised learning is to learn the classifier which predicts the labels of future examples as accurately as possible. Furthermore, a supervised learner can estimate the success, or the risk, of its hypotheses using the labeled training data by computing the empirical loss. Supervised learning algorithms aim to minimize some external error criterion, based on the difference between the targets and the outputs. Calculating and minimizing this error is possible because the target data is available. This is not the case in unsupervised learning. Targets are useful since they enable us to show the algorithm the correct answer to possible inputs. However, in many cases they are difficult to obtain, as in the present case with the manual annotation of mitochondria.

In *unsupervised* learning there are no targets available. The problem of unsupervised learning is also related to *manifold* learning and dimensionality reduction. The objective of this concept is to recover meaningful low-dimensional structures hidden in high-dimensional data. An example from the computer vision domain might be a set of pixel images of an individual's face observed under different pose and lighting conditions. The task here can be to identify the underlying variables (angle of elevation, direction of light, etc.) given only the high-dimensional pixel image data [192]. The classical techniques of manifold learning are metric multidimensional scaling, PCA and its nonlinear extension to kernel-PCA. Many more dimensionality reduction methods exist which consider linear or non-linear transformations and which are based on different concepts.

Clustering also belongs to the class of unsupervised machine learning methods and is one of the most widely used techniques for exploratory data analysis. Here, the objective is to organize

the data in some meaningful way. With clustering, however, there is no clear success evaluation procedure. Even with the knowledge of the underlying data distribution, it is not clear what is the correct clustering for that data or how to evaluate a proposed clustering [186]. A given set of objects can be clustered in various meaningful ways. This may be due to having different definitions of distance or similarity between objects. Generally, there may be several very different clustering solutions for a given dataset. As a result, there is a wide variety of clustering algorithms that, on the same input data, output very different results.

2.2.6.1 Clustering and Grouping

The hierarchical or linkage-based clustering algorithms proceed in a sequence of iterations as can be seen in Fig. 2.2.8. First, they start from the clustering that has each data point as a single-point cluster. Then, repeatedly, the algorithm merges the closest clusters of the previous clustering. Consequently, the number of clusters decreases with each such iteration. If kept going, such an algorithm would eventually result in the clustering in which all of the domain points share one large cluster. These types of algorithms require two parameters. The first considers the decision on how to measure or define the distance between clusters. The second parameter is needed to decide on when to stop merging the clusters. The input to a clustering algorithm is a between-points distance function d. The most common ways of extending d to a measure of distance between clusters $D(A, B)$ are single (2.29), average (2.30) and max (2.31) linkage distances:

$$D(A, B) = \min\{d(x, y) : x \in A, y \in B\} \tag{2.29}$$

$$D(A, B) = \frac{1}{|A||B|} \sum_{x \in A, y \in B} d(x, y) \tag{2.30}$$

$$D(A, B) = \max\{d(x, y) : x \in A, y \in B\} \tag{2.31}$$

A single linkage algorithm operates directly on a proximity matrix and is closely related to finding a minimal spanning tree on a weighted graph problem [186]. In order to obtain the partitioning of the space using dendrogram one needs to apply stopping criteria shown in Table 2.2.5.

Table 2.2.5: Stopping Criteria for Single Linkage Hierarchical Clustering.

Fixed number of clusters:	Fix some parameter k. Stop merging clusters as soon as the number of clusters is k
Distance upper bound:	Define a threshold r. Stop merging as soon as all the between-clusters distances are larger than r. It is also possible to set r to be $\alpha = \max\{d(x, y)\}$ for some $\alpha < 1$ [186]

SLHC (Single Linkage Hierarchical Clustering) has a property known as *chaining*. A few points that form a bridge between two clusters cause the single linkage clustering to unify these two clusters into one. This can sometimes be a disadvantage in some applications. The advantage of SLHC is, however, that it maintains good performance on data sets containing non-isotropic clusters, including well-separated, chain-like and concentric clusters. The main disadvantage of

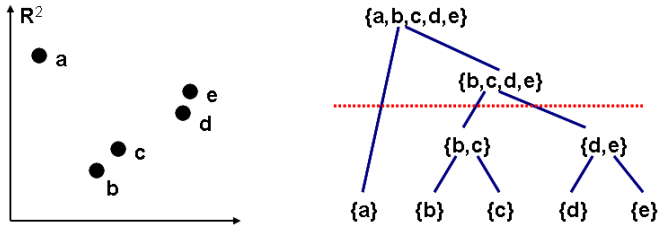


Figure 2.2.8: An example of a single linkage hierarchical clustering which uses the set of input elements $\mathcal{X} = \{a, b, c, d, e\} \subset \mathbb{R}^2$ and Euclidean distance as a distance measure in $\mathbb{R}^2$ (left) and produces the clustering dendrogram (right). A possible cut-off line (dashed red) is shown on the right. Image source: adapted from [186]. Diagrams are best viewed in color.

hierarchical methods is the inability to scale well. The time complexity of hierarchical algorithms is at least $O(m^2)$ (where m is the total number of instances), which is non-linear with the number of objects. Also, hierarchical methods do not have the back-tracking capability.

In contrast to hierarchical clustering, k-means clustering is a partitioning approach of the input space into mutually exclusive clusters. The k-means approach works with the Euclidean metric and aims at minimizing the sum-of-squares error from each datapoint to its cluster center. The algorithm starts by defining a cost function over a parameterized set of possible clusterings. The goal of the clustering algorithm is to find a partitioning of minimal cost. This is equivalent to an optimization problem where the objective function is a function from pairs consisting of an input and a proposed clustering solution. If we denote the objective function by G, the goal of the clustering algorithm is defined as finding for a given input $(\mathcal{X}, d)$ a clustering C so that $G((\mathcal{X}, d), C)$ is minimized [186].

The k-means objective function is one of the most popular clustering objectives. In k-means, the data are partitioned into disjoint sets $C_1, ..., C_k$ where each C_i is represented by a centroid μ_i. The k-means objective function measures the squared distance between each point in $\mathcal{X}$ to the centroid of its cluster. The centroid of C_i is defined as:

$$\mu_i(C_i) = \min_{\mu \in \mathcal{X}'} \sum_{x \in C_i} d(x, \mu)^2 \tag{2.32}$$

Then, the k-means objective can be written as:

$$G_{k-means}((\mathcal{X}, d), (C_1, ..., C_k)) = \min_{\mu_1, ..., \mu_k \in \mathcal{X}'} \sum_{i=1}^{k} \sum_{x \in C_i} d(x, \mu_i)^2 \tag{2.33}$$

In practical applications, finding the optimal k-means solution is often computationally infeasible [186]. An alternative is an iterative implementation which runs until the cluster centers stop moving [140]. In many cases, the term 'k-means clustering' refers to the outcome of the iterative algorithm rather than to the clustering that minimizes the k-means objective cost. Using the Euclidean distance function (2.34), that algorithm is summarized in Table 2.2.6.

$$d_i = \min_j d(\mathbf{x}_i, \boldsymbol{\mu}_j) \tag{2.34}$$

$$\boldsymbol{\mu}_j = \frac{1}{N_j} \sum_{i=1}^{N_j} \mathbf{x}_i \tag{2.35}$$

Table 2.2.6: The k-means Algorithm

1	Set a value for the number of clusters k, choose k random positions in the input space and assign the cluster centers $\boldsymbol{\mu}_j$ to these positions
2	For each datapoint $\mathbf{x}_i$ compute the distance to each cluster center $\boldsymbol{\mu}_j$ and assign the datapoint to the nearest cluster center with distance according to (2.34)
3	For each cluster center move the position of the center to the mean of the points in that cluster (N_j is the number of points in cluster j) according to (2.35)
4	Repeat until the cluster centers stop moving
5	For each test point compute the distance to each cluster center and assign the datapoint to the nearest cluster center with distance according to (2.34)

The k-means algorithm has two serious limitations. First the solution depends heavily on the initial positions of the cluster centers, resulting in poor minima, and second it can only find linearly separable clusters. In addition, k-means has problems when the data contain outliers and clusters have different sizes and densities. The authors in [209], for example, proposed a kernelized extension of the standard k-means algorithm that maps data points from input space to a higher dimensional feature space through a nonlinear transformation and minimizes the clustering error in feature space. Thus nonlinearly separated clusters are obtained in input space.

Many more clustering algorithms exist which specifically target the geometry of clusters, such as convex, non-convex or elongated clusters. Other algorithms consider such difficult problems as overlapping clusters, different densities and distributions, very large scale clustering and asymmetric datasets. A comprehensive review on various classical clustering algorithms is provided by Everitt et al. [64]. Since then, many new clustering concepts and formulations are continuing to emerge which are being inspired by different application domains. In Section 2.2.7.2 we review the SOM (Self-Organizing Map) algorithm which can be used to visualize overlapping clusters.

2.2.6.2 Dimensionality Reduction

Recent trends in machine learning see increasing interest in kernel methods, Bayesian reasoning, causality, information theoretic learning, reinforcement learning and nonnumeric data processing. In signal processing, Bayesian methods and graphical models are gaining popularity, while kernel approaches are still less prominent [149]. In machine learning, kernel-based learning has become a well-established technology within the last two decades. Kernel methods have enriched the spectrum of machine learning and statistical methods with a large number of nonlinear algorithms. In applications where the data have a large number of features, the dimensionality reduction finds a lower-dimensional representation preserving some of its properties. The key arguments for dimensionality reduction techniques are summarized in Table 2.2.7 [146].

Many linear scalar product-based algorithms can be kernelized with kernel-PCA, given that a positive definite kernel is used [149]. The so-called empirical kernel map allows processing of data by projecting it onto the leading kernel-PCA components. Thus, nonlinear variants of algorithms can be constructed via a nonlinear transformation. Such application areas as neuroscience, computational biology, natural language processing and physics have motivated the work, and these fields largely benefited from the novel set of tools. Kernel-based learning also enabled the addressing

Table 2.2.7: Key Arguments for Dimensionality Reduction.

Computational:	Compress the initial data as a preprocessing step to speed-up subsequent operations on the data. The curse of dimensionality implies that the higher number of dimensions requires more training data [140]
Visualization:	Visualize the data for exploratory analysis by mapping the input data into two- or three-dimensional spaces
Feature extraction:	Generate a smaller and more effective or useful set of features

of emerging industrial problems including social networks, text mining, and the general urge to understand better large and complex data. In computer vision, dimensionality reduction methods that are based on eigenvectors of the similarity matrix have also demonstrated good performance. These approaches are attractive because they are based on simple eigendecomposition algorithms whose stability is well-understood. Nevertheless, the use of eigendecomposition in the context of image segmentation is far from being a closed research topic [220]. Despite many empirical successes of spectral clustering methods, there are several unresolved issues. First, there is a wide variety of algorithms that use the eigenvectors in slightly different ways. Second, many of these algorithms have no proof that they will actually compute a reasonable clustering [155].

For VLS (Very Large Scale) problems such as pixel classification tasks, the computational cost of spectral decomposition quickly becomes prohibitive. This is amplified by the fact that in order to analyze the global saliency relationships, a dense similarity matrix which contains all pairwise similarity relations between all the pixels in the image, has to be constructed and processed. Sparsity in terms of reduced comparison space is often seen as a solution to speed-up computations. However, this is achieved at the cost of sacrificing global saliency information. An active field of theoretical research is the sub-sampling and low rank approximations of the similarity matrix. Some related methods are based on the highest column-wise second norm criterion and some are using eigenvalue and eigenvector estimation methods. Another topic of interest is concentrated on the development of parallel and distributed algorithms for spectral decomposition and processing of VLS similarity matrices. A list of dimensionality reduction algorithms which contains linear, non-linear, supervised and unsupervised formulations is provided in Table 2.2.8.

All the methods reviewed here are unsupervised except LDA. MDS, PCA and LDA are linear techniques. PCA is an unsupervised algorithm since it ignores class labels. In contrast to PCA, LDA is supervised and computes the directions (linear discriminants) that will represent the axes that maximize the separation between multiple classes. LDA finds the projection with maximum discrimination and is effective for classification. LDA is computed by solving a generalized eigenvalue problem. LDA is optimal when each class is Gaussian and has the same covariance matrix.

Classical kernel-PCA, Isomap and LLE algorithms can be applied to non-linear manifolds. Isomap generalizes MDS to non-linear manifolds, because it is based on replacing the Euclidean distance by an approximation of the geodesic distance on the manifold. LLE eliminates the need to estimate pairwise distances between widely separated data points. LLE recovers global non-linear

Table 2.2.8: Dimensionality Reduction Algorithms.

Algorithms	linear	non-linear	Neural Networks	spectral
MDS (Metric Multidimensional Scaling)	√			√
PCA (Principal Component Analysis)	√			√
FA (Factor Analysis)	√			√
ICA (Independent Component Analysis)	√		Neural-ICA	
LDA (Linear Discriminant Analysis)	√			√
kernel-PCA		√	Neural-PCA	√
Non-linear FA		√	MLP	
LLE (Locally Linear Embedding)		√	MLP	√
Isomap		√		√

structure from local linear fits. A parametric mapping between the observation and embedding spaces could be learned by supervised neural networks whose target values are separable by LLE.

Factor analysis is based on probabilistic modeling that is defined in terms of uncorrelated factors or latent variables. In contrast to PCA, there are no orthogonality constraints for the factors. The non-linear extension to FA can be implemented by the standard MLP network. The learning procedure is completely unsupervised because in non-linear FA only the outputs of the MLP network are known. More discussion on unsupervised neural networks is provided in Section 2.2.7.

Dimensionality reduction carried out either by PCA or kernel-PCA seeks to construct a feature space and to extract the significant orthogonal and uncorrelated principal components using the notion of the largest variance. Therefore, these methods are based on the assumption that the significant information is related to the eigenvectors which capture the highest variability in the data. The result of PCA and kernel-PCA is a set of orthogonal eigenvectors and a diagonal matrix containing eigenvalues ordered according to decreasing variance. PCA and metric MDS are simple spectral methods for linear dimensionality reduction. PCA is based on computing the low-dimensional representation of a high-dimensional dataset that preserves its covariance structure up to rotation. In PCA, the input patterns $x_i \in \mathbb{R}^d$ are projected into the m-dimensional subspace that minimizes the reconstruction error,

$$\epsilon = \sum_i \left\| x_i - \sum_{\alpha=1}^{m} (x_i \cdot e_\alpha) e_\alpha \right\|^2 \tag{2.36}$$

where the vectors $\{e_\alpha\}_{\alpha=1}^m$ define a partial orthonormal basis of the input space. It follows from (2.36) that the subspace with minimum ϵ is also the subspace with maximum variance. The basis vectors of this subspace are given by the leading m eigenvectors of the covariance matrix,

$$\mathbf{C} = \frac{1}{n} \sum_i x_i x_i^T \tag{2.37}$$

assuming that the input patterns x_i are centered at the origin. The outputs of PCA, denoted by ϕ_i, are the coordinates of the input patterns in this subspace, using the directions specified by these eigenvectors as the principle axes. Identifying e_α as the α^{th} top eigenvector of the covariance matric, the output $\phi_i \in \mathbb{R}^m$ for the input pattern $x_i \in \mathbb{R}^d$ has elements $\phi_{i\alpha} = x_i \cdot e_\alpha$. The eigenvalues of the covariance matrix in (2.37) measure the projected variance of the high-dimensional dataset

along the principal axes. Thus, the number of significant eigenvalues measures the dimensionality of the subspace that contains most of the data's variance. A large gap in the eigenvalue spectrum indicates that the data are mainly confined to a lower dimensional subspace.

2.2.6.3 Metric MDS

Metric MDS (MultiDimensional Scaling) is based on computing the low-dimensional representation of a high-dimensional dataset that preserves the inner products between different input patterns. Suppose we are given n objects, and for each pair (i, j) we have a measurement of the dissimilarity A_{ij} between the two objects. In MDS the aim is to place n points in a low-dimensional space (usually Euclidean) so that the dissimilarities are well-approximated by the interpoint distances d_{ij}. Let the coordinates of n points in p dimensions be denoted by $\mathbf{x}_i, i = 1, ..., n$, then $d_{ij}^2 = (\mathbf{x}_i - \mathbf{x}_j)^\top(\mathbf{x}_i - \mathbf{x}_j)$. In metric MDS the relationship is of the form $d_{ij} \approx f(A_{ij})$ where f is a specific analytic function, and for example polynomial transformations have been suggested [223]. The outputs $\phi_i \in \mathbb{R}^m$ of metric MDS are chosen to minimize the error function:

$$\epsilon = \sum_{ij} (x_i \cdot x_j - \phi_i \cdot \phi_j)^2 \tag{2.38}$$

The minimum error solution is obtained from the spectral decomposition of the Gram matrix $\mathbf{G}$ of inner products, $G_{ij} = x_i \cdot x_j$. Denoting the top m eigenvectors of this Gram matrix by $\{v_\alpha\}_{\alpha=1}^m$ and their respective eigenvalues by $\{\lambda_\alpha\}_{\alpha=1}^m$, the outputs of MDS are given by $\phi_{i\alpha} = \sqrt{\lambda_\alpha} v_{\alpha i}$.

MDS is motivated by the idea of preserving pairwise distances and is designed to preserve inner products. Let $\mathbf{A} : A_{ij} = \|x_i - x_j\|^2$ denote the matrix of squared pairwise distances (dissimilarities) between input patterns. MDS is often specified in this form. Assuming that the inputs are centered on the origin, a Gram matrix consistent with these squared distances is:

$$\mathbf{G} = -\frac{1}{2}(\mathbf{I} - \frac{1}{n}\mathbf{p}\mathbf{p}^T)\mathbf{A}(\mathbf{I} - \frac{1}{n}\mathbf{p}\mathbf{p}^T) \tag{2.39}$$

where $\mathbf{I}$ is the $n \times n$ identity matrix and $\mathbf{p} = (1, 1, ..., 1)^T$ is the uniform vector of length n. More details on MDS can be found in [41]. Metric MDS yields the same outputs ϕ_i as PCA – essentially a rotation of the inputs followed by the projection into the subspace with the highest variance. The outputs of both algorithms are invariant to global rotations of the input patterns. The Gram matrix of metric MDS has the same rank and eigenvalues up to a constant factor as the covariance matrix of PCA. If $\mathbf{X}$ denotes the $n \times n$ matrix of input patterns, then $\mathbf{C} = \frac{1}{n}\mathbf{X}\mathbf{X}^T$ and $\mathbf{G} = \mathbf{X}^T\mathbf{X}$. The equivalence of MDS and PCA follows from the singular value decomposition [223]. In both matrices, a large gap between the m^{th} and $(m+1)^{th}$ eigenvalues indicates that the high dimensional input patterns lie to a good approximation in a lower dimensional subspace of dimensionality m.

2.2.6.4 Kernel-PCA

Principal components analysis [101] is a classical method that provides a sequence of best linear approximations to a given high-dimensional observation. The subspace modeled by PCA captures the maximum variability in the data, and can be viewed as modeling the covariance structure of the data [77]. However, its effectiveness is limited by its global linearity. MDS, which is closely

related to PCA and for Euclidean distances produces the same results, has the same drawback. Suppose then we are given a real-valued function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ with the property that there exists a map $\Phi : \mathbb{R}^d \to \mathcal{K}$ into a dot product space $\mathcal{K}$ such that for all $(x, x') \in \mathbb{R}^d$, we have:

$$\Phi(x) \cdot \Phi(x') = k(x, x') \tag{2.40}$$

Examples of kernel functions that satisfy the above criteria include polynomial kernels $k(x, x') = (1 + x \cdot x')^p$ for positive integers p and Gaussian kernels $k(x, x') = \exp[-\|x - x'\|^2/\sigma^2]$. Many linear methods in statistical learning can be generalized to nonlinear settings with (2.40) by substituting the generalized dot products in feature space for Euclidean dot products in the input space [182].

Nonlinear generalization of PCA can be framed as kernel eigenvalue problems [181]. Given input patterns $(x_1, ..., x_n)$ where $x_i \in \mathbb{R}^d$, kernel-PCA computes the principal components of the feature vectors $(\Phi(x_1), ..., \Phi(x_n))$, where $\Phi(x_i) \in \mathcal{K}$. Since in general $\mathcal{K}$ may be infinite-dimensional, it is not possible to explicitly construct the covariance matrix in feature space. The problem has to be reformulated so that it can be solved in terms of the kernel function $k(x, x')$. Assuming that the data have zero mean in the feature space $\mathcal{K}$, its covariance matrix is given by:

$$\mathbf{C} = \frac{1}{n} \sum_{i=1}^{n} \Phi(x_i)\Phi(x_i)^T \tag{2.41}$$

The duality of PCA and MDS allows us to find the top eigenvectors of $\mathbf{C}$. Kernel-PCA can therefore be interpreted as a nonlinear version of MDS that results from substituting generalized dot products in feature space for Euclidean dot products in input space [223]. By using MDS, the top m eigenvalues and eigenvectors of the kernel matrix can be computed. The low-dimensional outputs ϕ_i of kernel-PCA are then given by $\phi_{i\alpha} = \sqrt{\lambda_\alpha} v_{\alpha i}$. In (2.41), it is assumed that the feature vectors have zero mean. In general, the mean $(1/n) \sum_i \Phi(x_i)$ has to be subtracted from each feature vector before computing $\mathbf{C}$ in (2.41). This leads to a different eigenvalue problem with:

$$\mathbf{G} = \mathbf{QCQ} \tag{2.42}$$

where $\mathbf{Q} = \mathbf{I} - \frac{1}{n}\mathbf{p}\mathbf{p}^T$. $\mathbf{G}$ is a centralized Gram matrix or a pseudo-covariance matrix. Additional constraints arise from the fact that the input data contained in the dissimilarity matrix $\mathbf{A}$ may not obey metric axioms [9] and therefore the intermediate step of Euclidean embedding becomes necessary [118, 176]. The embedding procedure attempts to construct a positive semidefinite $\tilde{\mathbf{G}}$ given generally indefinite $\mathbf{G}$. A particular method for Euclidean embedding, which is used throughout this work, is the Constant Shift Embedding introduced by Roth et al. [176]. Equation (2.43) shows five steps which are necessary in order to obtain the reconstructed feature vectors $\Phi(x_i)$ given that the similarity matrix $\mathbf{S}$ is not PSD and thus represents the underlying non-metric dataset.

$$x_i \to \mathbf{S} \to \mathbf{C} \to \mathbf{G} \to \tilde{\mathbf{G}} \to \tilde{\mathbf{U}}\tilde{\Lambda}\tilde{\mathbf{U}}^T \to \mathbf{\Phi} = \tilde{\mathbf{U}}\sqrt{\tilde{\Lambda}} \tag{2.43}$$

Transition $\mathbf{C} \to \mathbf{G}$ consists of the choice of the kernel function and the normalization procedure. Step $\mathbf{G} \to \tilde{\mathbf{G}}$ is optional with respect to the embedding into Euclidean space if the matrix $\mathbf{S}$ is indefinite. Step $\tilde{\mathbf{U}}\tilde{\Lambda}\tilde{\mathbf{U}}^T$ represents the spectral decomposition on the embedded $\tilde{\mathbf{G}}$, and the last step leads to the N-dimensional reconstructed feature space contained in the columns of $\mathbf{\Phi}$. The dimensionality reduction ($L < N$) is carried out by considering the first L column vectors of $\mathbf{\Phi}$.

2.2.7 Unsupervised Neural Networks

Unsupervised machine learning techniques can also be implemented using artificial neural networks. For instance, there are various neural algorithms which perform dimensionality reduction with neural-PCA and neural-ICA (Independent Component Analysis) models. The reason behind this is that using neural networks in unsupervised fashion can help to learn better representations of the input data.

In most supervised classification problems, finding and producing labels for the data is difficult and time consuming. In many application cases, plenty of unlabeled data exist and using them may improve the results. For example, a neural network can learn a mapping from an image to real-valued vector in such a way that resulting vectors are similar for images with similar content.

In most image classification tasks there are vastly more bits of information in the statistical structure of input images than in their labels [212]. In the context of deep learning, we have discussed the CNN developed by Krizhevsky et al. [112] in Section 2.2.2.3. Valpola [212] explained that in the ImageNet dataset with 1000 target classes, each label carries less than 10 bits of information. On the other hand, the amount of information contained in the 256 x 256 RGB images used as input, is several orders of magnitude more than 10 bits. This fact provides the motivation for research on deep unsupervised learning architectures.

The combination of the supervised and the unsupervised learning can be very efficient. The unsupervised learning aims to represent structure in the input data often by means of features. The resulting features can then be used as input for classification tasks or as initialization for further supervised learning. For example, Hinton et al. [93] proposed an unsupervised pre-training scheme which made subsequent supervised learning efficient for a deeper network than before. Valpola [212] has extended the above framework so that unsupervised learning can continue alongside supervised learning rather than be restricted to a preprocessing or a pre-training phase.

2.2.7.1 Neural-PCA and Neural-ICA

Kernel-PCA can also be framed as a neural networks model. Nonlinear neural-PCA is based on a MLP with an autoassociative topology, also known as an *autoencoder* or a bottleneck-type network. Such an autoencoder is trained to reconstruct original vectors from smaller representations (hidden layer activations) with reconstruction error as the cost function. This process creates meaningful low-dimensional representations of the input data that can be used for clustering. Scholz [183] explained neural-PCA with the five layers (3-4-1-4-3) MLP. This architecture is shown in Fig. 2.2.9 (left image) and consists of input $\mathbf{x}$ and output $\hat{\mathbf{x}}$ layers, two hidden layers (each with four nodes) and one component layer $\mathbf{z}$. The component layer in the middle is the bottleneck of the network because it has fewer units than in the input or output layers. Thus, the data have to be projected or compressed into a lower dimensional representation Z. The network consists of two major parts: the first part represents the extraction function $\Phi_{extr} : \mathcal{X} \rightarrow \mathcal{Z}$, whereas the second part represents the inverse function, the generation or reconstruction function $\Phi_{gen} : \mathcal{Z} \rightarrow \hat{\mathcal{X}}$. The hidden layers enable nonlinear mappings, and without them the network could only perform the linear Principal Component Analysis. Some other neural-PCA networks include the hierarchical, the circular and the inverse model for missing data [183].

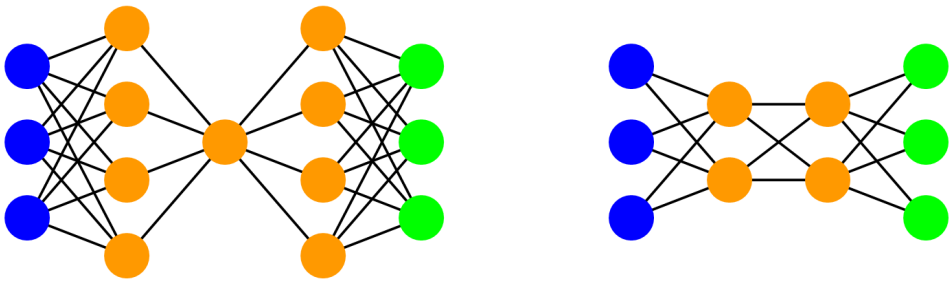


Figure 2.2.9: Left image shows a neural architecture of a bottleneck type autoassociative network (autoencoder) which can perform nonlinear PCA. The topology of this five-layer network is (3-4-1-4-3). The autoencoders are always symmetrical around the middle layer. Right image shows a neural network (3-2-2-3) for linear ICA. The second hidden layer is needed for data sphering. Image sources: Adapted from [183] and [150]. Diagrams are best viewed in color.

A linear neural-ICA algorithm based on information maximization, has been proposed, for example, by Bell and Sejnowski [16]. This algorithm did not assume any a priori knowledge of the input distributions and was designed for the zero noise limit. The proposed neural network was able to separate statistically independent components in the inputs. An example of a neural-ICA architecture is provided in Fig. 2.2.9 (right image). This feedforward neural network can perform Blind Source Separation and can provide the basis vectors of ICA as columns of the estimated mixing matrix. The second layer is optional if the data have to be *sphered* (or *whitened*) [150]. Whitening a matrix is one of the first steps in most ICA algorithms. A whitening transformation is a linear transformation that transforms a vector of random variables with a known covariance matrix into a set of new variables whose covariance is the identity matrix. This means that the new variables are uncorrelated and all have unity variance. This transformation is called 'whitening' because it changes the input vector into a white noise vector. This procedure also decorrelates all possible 'features' in the signal mixture. When the covariance matrix is identity matrix, the features are clustered in spheres[1] and therefore terms 'sphering' and 'whitening' are frequently interchangeable in the literature. According to Bell and Sejnowski [16], the information maximization-based neural-ICA network described is limited in discovering nonlinear optimal mappings, while the addition of more hidden layers could be more powerful.

Nonlinear ICA is a more difficult task because it depends on the a priori knowledge of the nonlinear mixing process. Therefore, special nonlinear ICA models simplify the problem to particular applications in which some information about the mixing system and the source signals is available. A review of many ICA algorithms and applications is given, for example, by Hyvärinen and Oja [96], and the mathematics of ICA is provided in Section 3.3.1.

2.2.7.2 Self-Organizing Maps and Neural k-means

There are also a number of different clustering algorithms based on neural networks. The most widely known one is the Self-Organizing Map or Kohonen's networks [171], which was proposed by Teuvo Kohonen in 1988. An example of this type of neural network in provided in Fig. 2.2.10

[1] See Section 3.3.1 for more detailed explanation.

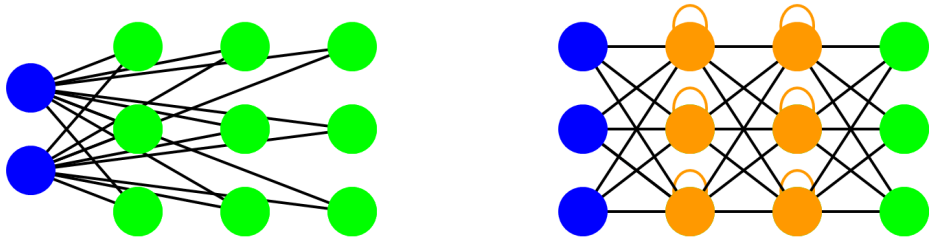


Figure 2.2.10: Left image shows an example of a Self-Organizing Map neural network. Right image shows a Recurrent Neural Network. The two middle layers contain neurons with feedback connections. Image sources: Adapted from [214]. Diagrams are best viewed in color.

(left image) and contains two layers. The input layer consists of p-dimensional observations $\mathbf{x}$. The output layer consists of k nodes for the k clusters, each of which is associated with a p-dimensional weight $\mathbf{w}$ [64]. This is a competitive-type learning neural network that has a set of neurons connected to form a topological grid. When some pattern is presented to the SOM, the neuron with the closest weight vector is considered a winner and its weights are adapted to the pattern, as well as the weights of its neighborhood. In this way, the SOM naturally finds data clusters. The ability to self-organize facilitates adaptation to previously unknown input data. This process appears to be a natural way of neural biological learning where the patterns take shape during the learning process. The other feature of self-organization is that a global ordering of the input space is achieved by using only a set of local interactions. This is based on the fact that the neurons that are very far apart do not interact with each other.

Various extensions of SOMs have been proposed since 1988. One strategy was to apply a competitive learning rule that minimizes the mean absolute error between the input samples and the weight vectors. Here an information-theoretic criterion can be optimized directly, instead of minimizing a distortion criterion. In the context of topographic map formation, Ralph Linsker was among the first to explore this idea. He studied self-organization in a perceptual network and proposed a principle of maximum information preservation [126] according to which a processing stage has the property that the output signals will optimally discriminate, in an information-theoretic sense, among possible sets of input signals applied to that stage. This algorithm is also referred to as *infomax*. Furthermore, Linsker developed a learning rule for topographic map formation in a probabilistic network by maximizing the average mutual information between the output and the signal part of the input which was corrupted by noise [127].

In order to overcome the topology mismatches that occur with the original SOM algorithm, as well as to achieve an optimal use of the neurons, the geometry of the lattice has to match that of the data manifold it is intended to represent [213]. For that purpose, several incremental or structure-adaptive *growing* self-organizing map algorithms have been developed. They all share one property where the lattices do not have a pre-defined structure, but are rather gradually build up. Therefore the number of neurons and the lattice dimensionality can vary. The lattice is generated by a successive insertion and an occasional deletion of neurons and connections between them.

The other family of SOM algorithms, which is relevant for this thesis, is Kernel Topographic Maps. In the SOM algorithm, topographic maps have disjoint and uniform activation regions. In order to accommodate neurons with overlapping activation regions, the kernel topographic maps

include various kernel functions. Such algorithms model the input density with a kernel mixture. The use of kernels has biological relevance and is seen to improve density estimation properties of topographic maps. The advantage is that the unique visualization properties of topographic maps are combined with an improved modeling of clusters in the data. The first example of kernel-based topographic maps is the elastic net developed by Durbin and Willshaw [61]. The elastic net has found application in a number of areas. Some examples include finding trajectories of charged particles with multiple scattering in high energy physics experiments and the prediction of the protein folding structure. It has been also used for clustering applications. In computer vision, the elastic net has a close relationship with "snakes". This algorithm has been, for example, used in magnetic resonance imaging for finding lung boundaries and extracting the shape of a closed object [213]. Another kernel-based SOM is the Generative Topographic Map algorithm, introduced by Bishop et al. [21, 22]. This algorithm allows general non-linear transformations from latent space to data space. This algorithm develops a topographic map that attempts to find a representation for the input distribution in terms of a number of latent variables. Generative topographic mapping has found many real-world applications. Some examples include visualization of oil flows along multi-phase pipelines, visualization of electropalatographic data for investigating the activity of the tongue in normal and pathological speech. Other applications include classification of in vivo magnetic resonance spectra of controls and Parkinson patients, word grouping in document data sets, the exploratory analysis of web navigation sequences and spatiotemporal clustering of transition states of a typhoon from image sequences of cloud patterns. Another application is the micro-array analysis of gene expression data with the purpose of finding low-confidence value genes [213].

Finally, the neural k-means clustering algorithm is another example of competitive learning. In this formulation, the neurons compete with each other to fire and the winning neuron is the one that best matches the input. Marsland [140] showed how a neural network can implement the k-means solution. A set of neurons can be used to imitate the k-means algorithm which, for example, consists of one layer of neurons, some input nodes and no bias node. The first layer contains the inputs, which do not do any computation and the second layer is a layer of competitive neurons.

2.2.7.3 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) is another class of deep networks for unsupervised learning, where the depth can be as large as the input sequence. The RNNs are used to predict a data sequence in the future using previous data samples, and no additional class information is used for learning. In the artificial RNN, as illustrated in Fig. 2.2.10 (right image), neurons are fed information, not just from the previous layer, but also from themselves from the previous time step. The concept is based on the biological evidence that the human brain is a recurrent neural network. Artificial RNNs are very powerful for modeling sequence data, but are difficult to train to capture long-term dependencies [46]. Nevertheless, it is currently acknowledged that RNNs are more powerful and biologically more plausible than other adaptive approaches such as Hidden Markov Models, feedforward neural networks and Support Vector Machines [180].

RNNs have been also applied to the BSS problem which is usually done with ICA. Authors in [8], for example, reported that their developed RNN needed a much smaller operational range

for the synaptic weights. Therefore, the hardware requirement for the RNN was less that the one for the usual feedforward neural network. Stephen Grossberg of Boston University has done many studies on the modeling of human visual system with a system of neurons and has compiled a comprehensive summary on biological RNNs that are found in the brain [82].

Another unsupervised neural network architecture is based on the Adaptive Resonance Theory (ART) developed by Carpenter and Grossberg [29]. This theory of human cognitive information processing has inspired many neural models for pattern recognition and unsupervised learning. ART systems have been used to explain a variety of cognitive and neurobiological data. The adaptive ART neural networks self-organize a stable pattern recognition code in real-time in response to arbitrary input patterns sequences. ART principles have further helped explain parametric behavioral and brain data in the areas of visual perception and object recognition. A combination of an ART and a RNN which adaptively learns an input-output mapping using both supervised and unsupervised formulations is presented, for example, by Vieira and Lee in [215]. An application of a multidimensional RNN to the segmentation of neuronal structures in three-dimensional Electron Microscopy images is given by Stollenga et al. [198].

In summary, this chapter provides a comprehensive survey on traditional and emerging machine learning techniques and their applications to biomedical imaging. We conclude that the state of the art in cellular and subcellular image segmentation consists often of both supervised and unsupervised machine learning formulations. We covered the relevant mathematical foundations of dimensionality reduction, kernels and clustering because they are used in the subsequent contribution-related Chapters 4, 5 and 6. In particular, we apply embedded kernel-PCA, k-means and single linkage hierarchical clustering to solve the problems of localization and segmentation of mitochondria in electron microscopy images. In addition, this thesis is set within the model-based spectral clustering framework and provides also an alternative solution to the eigendecomposition-based spectral clustering algorithms. In the next chapter, we review the graph-based spectral clustering algorithms, alternative and combined dimensionality reduction approaches, solutions to the very large scale spectral clustering, and discuss the importance of perceptual grouping and its potential for biomedical image segmentation.

Chapter 3

Model-based Spectral Clustering

We have reviewed the foundations of machine learning in the previous chapter. In particular, an unsupervised learning approach to the segmentation of mitochondria has been applied in the research developed in this thesis. Spectral clustering combines the two principle elements of unsupervised machine learning: dimensionality reduction and clustering in the projected feature space. In general, spectral clustering is concerned with the clustering of points using eigenvectors of model-based similarity matrices derived from the data. In these approaches the data are firstly mapped to a low-dimensional space where it can be separated and be easily clustered. We have reviewed the spectral dimensionality reduction methods in the previous sections and in this thesis we are particularly interested in kernel-PCA which computes the eigenvectors of the normalized covariance matrix. There are also many different algorithms on spectral clustering which differ in the ways that the similarity or dissimilarity matrices are constructed and normalized, in how the eigenvectors are computed, and by which (largest or smallest) and how many eigenvectors are used for clustering.

3.1 Graph-Based Spectral Clustering Algorithms

Below we review some selected algorithms with links to spectral graph partitioning which aim to identify a graph partition such that the edges between different groups have low weights and the edges within a group have high weights. Good reviews of these approaches are given in [155, 190] and [134]. An important notion in graph-based spectral clustering is graph Laplacian matrices. There exists a whole field dedicated to the study of those matrices, called spectral graph theory (e.g., see [36]) and there are many different definitions of graph Laplacians. The *unnormalized* graph Laplacian is the square $m \times m$ matrix $L = D - W$ where D is a diagonal matrix with elements:

$$D_{i,i} = \sum_{j=1}^{m} W_{i,j} \tag{3.1}$$

and where $W_{i,j} = s(x_i, x_j)$ is the weight of the graph edge, and is basically the similarity $s(x_i, x_j)$ between two graph vertices x_i and x_j representing two data points. The matrix D is called the degree matrix. Let $C_1, ..., C_k$ be a clustering and let $H \in \mathbb{R}^{m,k}$ be the matrix [186] such that

$$H_{i,j} = \frac{1}{\sqrt{|C_j|}} \mathbf{1}_{[i \in C_j]} \tag{3.2}$$

Then, the columns of H are orthonormal to each other and

$$RatioCut(C_1, ..., C_k) := \text{trace}(H^\top LH) \tag{3.3}$$

Therefore, to minimize RatioCut we can search for a matrix H whose columns are orthonormal and such that each $H_{i,j}$ is either 0 or $1/\sqrt{|C_j|}$. This is, however, an integer programming problem which cannot be solved efficiently. Instead, we can relax the latter requirement and search for an orthonormal matrix $H \in \mathbb{R}^{m,k}$ that minimizes $\text{trace}(H^\top LH)$. The unnormalized spectral clustering algorithm therefore starts with finding the matrix H of the k-eigenvectors corresponding to the smallest eigenvalues of the graph Laplacian matrix.

Table 3.1.1: Unnormalized Spectral Clustering Algorithm.

1	Construct similarity matrix $S \in \mathbb{R}^{m \times m}$ and set the number k of clusters
2	Construct a similarity graph. Let W be its weighted adjacency matrix
3	Compute the unnormalized graph Laplacian L
4	Let $U \in \mathbb{R}^{m,k}$ be the matrix whose columns are the eigenvectors of L corresponding to the k smallest eigenvalues
5	Let $\mathbf{v}_1, ..., \mathbf{v}_m$ be the rows of U
6	Cluster the points $\mathbf{v}_1, ..., \mathbf{v}_m$ using k-means
7	Output clusters $C_1, ..., C_K$ of the k-means algorithm

There are two matrices which are called normalized graph Laplacians in the literature. Both matrices are closely related to each other and are defined as:

$$L_{sym} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2} \tag{3.4}$$

$$L_{rw} = D^{-1} L = I - D^{-1} W \tag{3.5}$$

We denote the first matrix by L_{sym} as it is a symmetric matrix, and the second one by L_{rw} as it is closely related to a random walk. There are two different versions of normalized spectral clustering, depending which of the normalized graph Laplacians is used.

Table 3.1.2: Normalized Spectral Clustering Algorithm according to Shi and Malik [190]

1	Construct similarity matrix $S \in \mathbb{R}^{n \times n}$ and set the number k of clusters
2	Construct a similarity graph. Let W be its weighted adjacency matrix
3	Compute the unnormalized Laplacian L
4	Compute the first k generalized eigenvectors $u_1, ..., u_k$ of the generalized eigenproblem $Lu = \lambda Du$
5	Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, ..., u_k$ as columns
6	For $i = 1, ..., n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U
7	Cluster the points $(y_i)_{i=1,...,n}$ in $\mathbb{R}_k$ with the k-means algorithm into clusters $C_1, ..., C_k$
8	Output clusters $A_1, ..., A_k$ with $A_i = \{j \vert y_j \in C_i\}$

The above Shi and Malik algorithm uses the generalized eigenvectors of L, which correspond to the eigenvectors of the matrix L_{rw} [134]. So in fact, the algorithm works with eigenvectors of the normalized Laplacian L_{rw}, and hence is called normalized spectral clustering.

The next algorithm, developed by Ng et al. [155] also uses a normalized Laplacian, but this time uses the matrix L_{sym} instead of L_{rw}. This algorithm introduces an additional row normalization in step 6 which is not needed in the other algorithms.

Table 3.1.3: Normalized Spectral Clustering Algorithm according to Ng, Jordan and Weiss [155]

1	Construct similarity matrix $S \in \mathbb{R}^{n \times n}$ and set the number k of clusters
2	Construct a similarity graph. Let W be its weighted adjacency matrix
3	Compute the normalized Laplacian L_{sym}
4	Compute the first k eigenvectors $u_1, ..., u_k$ of L_{sym}
5	Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, ..., u_k$ as columns
6	Form the matrix $T \in \mathbb{R}^{n \times k}$ from U by normalizing the rows to norm 1, that is set $t_{ij} = u_{ij} / (\sum_k u_{ik}^2)^{1/2}$
7	For $i = 1, ..., n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of T
8	Cluster the points $(y_i)_{i=1,...,n}$ in $\mathbb{R}_k$ with the k-means algorithm into clusters $C_1, ..., C_k$
9	Output clusters $A_1, ..., A_k$ with $A_i = \{j \vert y_j \in C_i\}$

All three algorithms stated above look quite similar, apart from the fact that they use three different graph Laplacians. Experimentally it has been observed that using more eigenvectors and directly computing a k-way partitioning gives better clustering results [155, 5, 136]. In their original paper, Ng et al. construct the similarity (affinity) matrix S by using the Gaussian model:

$$S_{ij} = \exp(-\|s_i - s_j\|^2 / 2\sigma^2) \tag{3.6}$$

if $i \neq j$ and $S_{ii} = 0$. Here the scaling parameter σ^2 controls how rapidly the affinity falls off with the distance between s_i and s_j and the authors show how to tune that parameter automatically.

A simple spectral clustering algorithm which does not use the graph Laplacian matrices, but rather directly operates on the constructed similarity matrix is given by Perona and Freeman in [166]. The authors have observed a new property of the scene by calculating the pointwise eigenvector-based approximation p_i of the pairwise affinity S_{ij} of the elements in the scene. This global property describes the partition of a scene into foreground and background where p can be seen as a saliency function of the points. After discovering the concept of foreground, the authors specify the algorithm (see Table 3.1.4) for calculating the foreground group.

Table 3.1.4: Affinity Factorization Algorithm according to Perona and Freeman in [166]

1	Form a matrix S_{ij} containing the pairwise affinity of each pair of elements in the scene
2	Call p the eigenvector of S that is associated to its largest eigenvalue
3	Define the foreground F as the set of objects i whose corresponding p_i is not equal to zero

It is important to note at this stage that all four graph-based spectral clustering approaches reviewed use a similarity (sometimes also called affinity) matrix as an input. By way of comparison, spectral dimensionality reduction methods such as MDS, PCA and kernel-PCA require a dissimilarity (pairwise distances) measure as an input. Therefore, in the latter case one has to apply a dissimilarity mapping on the defined similarity model.

The advantages of spectral clustering are that this method does not make strong assumptions about the statistics of the clusters, it is easy to implement by means of linear algebra, it provides good clustering results and it has been proven to capture the *perceptual* organization of data.

We review work on visual perception and similarity models for perceptual grouping in Section 3.4.1. Spectral clustering is also reasonably fast for sparse datasets of several thousand elements. The disadvantages include the sensitivity to the choice of model parameters and the fact that spectral clustering may quickly become computationally expensive for large datasets such as for example high-resolution natural images.

3.2 Related Work on Large Scale Spectral Clustering

Because for natural images of size $N \times M$ the similarity matrix acquires a size of $NM \times NM$, clustering approaches which rely on the direct analysis of affinity or similarity matrices are infeasible for large scale problems (large matrices). We address this computational limitation by deriving a new anticorrelation-based formulation of spectral clustering in Chapter 6 [52]. In order to understand our motivation and strategy for this work, we first review related publications on large scale spectral clustering in the following sections.

We start with the work by Donoser et al. [54] where the authors noted that for efficiency reasons they split the image into non-overlapping equal-sized blocks. In each of these blocks they independently clustered the pixels based on their 36-dimensional feature space by applying affinity propagation clustering [70] to the affinity matrix. Furthermore, they merged the local solutions obtained based on clustering the mean feature vectors (of each block then), again by affinity propagation. Mahamud et al. [135] noted that ordinary techniques for the computation of eigenvectors and eigenvalues are infeasible for large images. In an attempt to solve this problem, the authors developed a technique which exploits the sparseness and symmetry of the similarity matrix and claimed to be able to reduce significantly the time required to compute the eigendecomposition.

For the case of using spectral clustering methods on natural images, the cost quickly becomes prohibitively high and makes spectral methods impractical [102]. In this regard, the problem of extracting the eigenvectors and eigenvalues of large similarity matrices is approached from the perspective of either estimation or distributed computing. A relatively recent work towards a practical implementation strategy is the work on parallel spectral clustering in distributed systems where the authors design a parallel and distributed spectral clustering algorithm [35]. This work also shows that the problem is still open for a theoretical solution. The most commonly used approach to address the computational and memory difficulties is to zero out some elements in the similarity matrix, i.e., to sparsify the matrix. From the sparse similarity matrix obtained, one then finds the corresponding Laplacian matrix and calls a sparse eigensolver. Several methods are available for sparsifying the similarity matrix [134].

A sparse representation effectively handles the memory bottleneck, but some sparsification schemes still require calculating all elements of the similarity matrix. An example here is provided by the work of Crum [42] on voxel classification in 3D brain MRI images. In order to sparsify the similarity matrix, the author adopted a stochastic sparse sampling approach where the similarity matrix $\mathbf{S}$ is sparsily filled by computing pairwise similarities between each point and m other randomly selected points. The sparse eigenvector decomposition for computing the first $k+1$ eigenvalues is achieved with the Jacobi-Davidson method and the author further noted that the stochastic sampling process affects the quality of classification.

Another popular approach to speed up spectral clustering is to use a dense sub-matrix of the similarity matrix by using for example the Nyström approximation. In their work, Chen et al. [35], for example, aimed at developing a parallel spectral clustering package on distributed environments. For that reason, the authors presented a comparison between the traditional method of sparsifying the similarity matrix and the Nyström method. The authors also highlighted that the main bottleneck is in computation of similarity matrix. In this regard, Liu et al. [129] also stated that the majority of the time is actually spent on constructing the pairwise distance and affinity matrices. Comparatively, the time spent on actual clustering is almost negligible.

Low rank approximations have recently gained broad popularity in computer science. For example, in areas such as computer vision, information retrieval and machine learning, they are used as a basic tool for extracting correlations and removing noise from matrix structured data. However, application of this technique to massive matrices quickly runs against practical computational limits. Specifically, orthogonal iteration and Lanczos iteration, the two most common algorithms for computing low rank approximations, operate through repeated matrix-vector multiplication, thereby requiring superlinear time and large working sets [80].

When only a few eigenvectors and eigenvalues are required, there exist less computationally intensive techniques such as the Jacobi, the Arnoldi and the more recent Hebbian methods [80, 81]. These iterative methods also require computation of matrix-vector products at each step and involve several passes through the data. When the matrix is sparse, these techniques can be implemented relatively efficiently. However, when dealing with a large dense matrix these products become expensive to compute. In this regard, Talwalkar et al. [201], for example, reported that when working with 18 million data points, it is not possible even to store the full matrix (~1600TB), rendering the iterative methods infeasible. Belabbas and Wolfe [15] also stressed that for the growing number of applications dealing with very large high-dimensional datasets, the optimal approximation afforded by an exact spectral decomposition is too costly, because its complexity cost is the cube of either the number of training examples or their dimensionality.

Motivated by such applications, a number of approaches to obtaining alternative low-rank decompositions have been applied in the statistical machine learning literature, many of them relying on the Nyström method to approximate a positive definite kernel. The Nyström method belongs to the class of random sampling techniques that provide a powerful alternative for approximate spectral decomposition and only operate on a subset of matrix. The Nyström approximation has been primarily studied in the machine learning community [57, 224]. An alternative Column-sampling technique has been analyzed in the theoretical computer science community [48]. A spectral clustering formulation for very large scale problems based on the Nyström method has been presented, for example, by Fowlkes et al. [68] where the authors avoided calculating the whole similarity matrix. This approach traded accurate similarity values for shortened computational time.

Generally, in order to approach the computational limitation of spectral clustering there are currently three major strategies: sparsification of the similarity matrix, low rank matrix approximation [201] and the Nyström method, which due to its popularity is reviewed in detail next.

3.2.1 Nyström and Column-sampling Methods

Historically, the Nyström extension was introduced to obtain numerical solutions to integral equations [15]. Let $g : [0,1] \times [0,1] \rightarrow \mathbb{R}$ be a Symmetric Positive SemiDefinite kernel and $(u_i, \lambda_i^u), i \in \mathbb{N}$, denote its pairs of eigenfunctions and eigenvalues as follows:

$$\int_0^1 g(x,y)u_i(y)dy = \lambda_i^u u_i(x), i \in \mathbb{N} \tag{3.7}$$

Eigenfunction notation refers to the continuous form of eigenvectors. The Nyström extension provides a means of approximating k eigenvectors of $g(x,y)$ based on an evaluation of the kernel at k^2 distinct points $\{(x_m, x_n)\}_{m,n=1}^k$ in the interval $[0,1] \times [0,1]$. Defining a kernel matrix $\mathbf{G}(m,n) \equiv \mathbf{G}_{mn} := g(x_m, x_n)$ composed of these evaluations leads to the m coupled eigenvalue problems:

$$\frac{1}{k}\sum_{n=1}^{k} \mathbf{G}(m,n)u_i(n) = \lambda_i^u u_i(m), i = 1,2,...,k \tag{3.8}$$

where (u_i, λ_i^u) represent the k eigenvector-eigenvalues pairs associated with $\mathbf{G}$. These pairs may then be used to form an approximation $\tilde{u_i} \cong u_i$ to the eigenfunctions of g as follows:

$$\tilde{u_i}(x) = \frac{1}{\lambda_i^u k}\sum_{m=1}^{k} g(x, x_m)u_i(m) \tag{3.9}$$

The essence of the method is thus to use only partial information about the kernel to first solve a simpler eigenvalue problem, and then to extend the eigenvectors obtained therewith by using complete knowledge of the kernel. Specifically, one may approximate k eigenvectors of $\mathbf{G}$ by decomposing and then extending a $k \times k$ principal submatrix of $\mathbf{G}$. Let us use Nyström submatrix notations $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{Z}$ in this Section. Let $\mathbf{G}$ be partitioned as:

$$\mathbf{G} = \begin{pmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{C} \end{pmatrix}, \mathbf{Z} = \begin{pmatrix} \mathbf{A} \\ \mathbf{B} \end{pmatrix} \tag{3.10}$$

With $\mathbf{A} \in \mathbb{R}^{k \times k}$ we define two spectral decompositions: $\mathbf{G} = \mathbf{U}\Lambda\mathbf{U}^T$ and $\mathbf{A} = \mathbf{U}_A\Lambda_A\mathbf{U}_A^T$. The Nyström extension then provides an approximation for k eigenvectors in $\mathbf{U}$ as:

$$\tilde{\mathbf{U}} = \begin{pmatrix} \mathbf{U}_A \\ \mathbf{B}\mathbf{U}_A\Lambda_A^{-1} \end{pmatrix} \tag{3.11}$$

The above result in given by Belabbas and Wolfe [15] without derivation. The approximations $\tilde{\mathbf{U}} \cong \mathbf{U}$ and $\Lambda_A \cong \Lambda$ may be composed to yield an approximation $\tilde{\mathbf{G}} \cong \mathbf{G}$ according to:

$$\tilde{\mathbf{G}} := \tilde{\mathbf{U}}\Lambda_A\tilde{\mathbf{U}}^T = \begin{pmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T \end{pmatrix} = \mathbf{Z}\mathbf{A}^{-1}\mathbf{Z}^T \tag{3.12}$$

In the above formulation, $\tilde{\mathbf{G}}$ is called the Nyström approximation of $\mathbf{G}$. Equation (3.11) shows that the main computational burden now takes place on a principal submatrix $\mathbf{A}$ of dimension $k < n$, and hence the Nyström extension provides a practical means of scaling up spectral methods in machine learning to very large kernels. From (3.10) and (3.12) it can be deduced that the resultant approximation error is:

$$\|\mathbf{G} - \tilde{\mathbf{G}}\| = \|\mathbf{C} - \mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T\|, \tag{3.13}$$

where $Sh^{(c)}(\mathbf{A}) := \mathbf{C} - \mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$ is known as the *Schur* complement of $\mathbf{A}$ in $\mathbf{G}$. The equation in (3.13) ties the quality of the Nyström approximation explicitly to the partitioning of $\mathbf{G}$. Intuitively, this error reflects the loss of information that results from discarding submatrix $\mathbf{C}$ while retaining $\mathbf{A}$ and $\mathbf{B}$. The Nyström method yields a means of approximating $\mathbf{G}$ conditioned on a particular choice of partition, hence shifting the computational load to determining that partition. To calculate approximations to the top k eigenvectors and eigenvalues of $\mathbf{G}$, the runtime of this method is $O(m^3 + kmn)$, that is, m^3 for eigendecomposition of $\mathbf{A}$ and kmn for multiplication with $\mathbf{B}$.

In contrast to the Nyström method, the Column-sampling method [201] uses the direct decomposition of the matrix $\mathbf{Z}$. More detailed, it approximates eigen-decomposition of $\mathbf{G}$ by using the SVD of $\mathbf{Z}$ directly. In equation (3.12) the Column-sampling method replaces $\mathbf{A}^{-1}$ with $\sqrt{\frac{m}{n}}(\mathbf{Z}^T\mathbf{Z})^{\frac{1}{2}}$ [201]. The SVD on $\mathbf{Z}$ has a cost of $O(nm^2)$, but since it can not be easily parallelized, it is still quite expensive when $n \geq 18$ million as in the study conducted by Talwalkar et al. [201]. Generally, the time needed to calculate approximations to the top k eigenvectors and eigenvalues of $\mathbf{G}$ is $O(nm^2 + m^3)$, that is, nm^2 to generate $\mathbf{Z}^T\mathbf{Z}$ and m^3 for SVD on $\mathbf{Z}^T\mathbf{Z}$. In addition, the authors also investigated how the two described approximation techniques differ in their treatment of negative eigenvalues and the corresponding eigenvectors. The authors concluded that the Nyström method allows one to use eigenvalue decomposition of $\mathbf{A}$ to yield signed eigenvalues, making it possible to discard the negative eigenvalues and the corresponding eigenvectors. On the contrary, it is not possible to discard these in the Column-based method, since the signs of the eigenvalues are lost in the SVD of the rectangular matrix $\mathbf{Z}$ (or the eigenvalue decomposition of $\mathbf{Z}^T\mathbf{Z}$).

3.2.2 Low-rank Approximations

There is an ongoing research on the optimal low rank approximations of large matrices in terms of random sampling theory. As pointed out by Talwalkar et al. [201], in the Nyström method, the matrix $\mathbf{A}$ is usually constructed by randomly sampling m columns of $\mathbf{G}$, where $m \ll n$. Other sampling schemes have been suggested which consider the optimal selection of optimal submatrix $\mathbf{A}$ from all possible partitions of $\mathbf{G}$. Karoui and d'Aspremont [102] for instance, proposed a randomized, distributed algorithm to estimate eigenvectors and eigenvalues which makes spectral methods tractable on very large scale matrices. In fact, their approach falls into the category of subsampling of similarity matrices in order to lower the complexity of spectral methods. The idea is to find a good approximation of a matrix, $\mathbf{G}$ that has low rank. Achlioptas and McSherry [1] introduced a technique for accelerating the computation of such approximations when $\mathbf{G}$ has strong spectral features, i.e. when the singular values of interest are significantly greater than those of a random matrix with size and entries similar to $\mathbf{G}$.

The first mathematically rigorous approach to speeding up the computation of low rank approximations by employing randomization was given in the work of Frieze, Kannan and Vempala [71], where it was shown that one can efficiently approximate $\mathbf{G}$ from a submatrix whose size is independent of m, n. Specifically, the authors showed how to compute a rank k matrix $\mathbf{A}$ such that

$$\|\mathbf{G} - \mathbf{A}\|_F^2 \leq \|\mathbf{G} - \mathbf{G}_k\|_F^2 + \epsilon\|\mathbf{G}\|_F^2 \tag{3.14}$$

from a square submatrix of $\mathbf{G}$ having dimension $d \geq 10^7 k^4/\epsilon^3$. The above approach was extended in subsequent works of Drineas et al. [55, 56] motivated by practical considerations. Specifically,

by sampling entire columns of **G**, the algorithms in [55, 56] greatly improve the dependence of d on k and ϵ at the cost of introducing a linear dependence of d on $m \leq n$.

Instead of sampling random submatrices of **G**, Achlioptas and McSherry [1] independently sampled individual entries of **G**. By breaking the correlation between the elements of the same row/column in the sampling process, the authors gained access to a wealth of results from the theory of random matrices. These enable the derivation of the very sharp matrix perturbation bounds. Moreover, this entry-wise independence allows one to perform the sampling in a single pass over the matrix. In contrast, the correlation among row/columns requires the algorithms in [55, 56, 71] to make two passes over the matrix **G**: the first to sample a set of indices and the second to collect the contents of the corresponding submatrix. Such a second pass can be impractical or even impossible for massive data sets, where one often has only streaming access to the data [1].

3.3 Modifications and Alternatives to Spectral Methods

Practical spectral clustering approaches to biomedical image segmentation usually resolve the complexity problem of the input data by using block-processing, downsampling, compression or the concept of superpixels. Also, emerging and more efficient new hardware platforms as well as distributed and parallel computing architectures allow one to deal with this complexity problem more effectively. In addition, a set of extracted low-level features, for example, may be much smaller than the number of pixels in the original image. Usually, one can expect the number of low-level primitives, such as contours or line segments to be in the order of 100s or 1000s. By considering pixel classification tasks, the number of features to analyze becomes astronomical, thus revealing the bottleneck of spectral clustering. Therefore, the main motivation for our work which we present in Chapter 6 comes from the established fact that presently available spectral clustering algorithms cannot directly be applied to general very large scale problems. One example at this point can be given by the work of Crum [42] where the author presented a case study on tissue classification in brain MRI images. MRI volumes are typically of size $n = 256 \times 256 \times 128 \sim 8$ million voxels total. The computational problems are immediately apparent: first the number of voxel similarity comparisons in the feature generation stage is $n(n-1)/2$, and second, an eigen-analysis of an $n \times n$ symmetric matrix is required. When number of voxels $n >\sim 1$ million as for three-dimensional MRI brain images this calculation is impractical [42].

Anther factor, relating to the development of possible theoretical alternatives to very large scale spectral clustering in this work, comes from the hypothesis that the information about the cluster arrangements in the feature space is already contained in the similarity matrix (though maybe in a different form), and therefore can potentially be extracted ahead of the spectral decomposition. Weiss [220], for example, has noted that "...from visual inspection, the affinity matrix contains information about the correct segmentation". Bie and Cristianini [20] noted that the kernel matrix contains sufficient information to run many classical and new linear algorithms in the embedding space. The question which naturally arises then is whether an eigenvector is the only conveyor of the information necessary to achieve the correct clustering? From a separate direction determined by research on ICA also comes the idea that physically meaningful signals underlying a data set should or could be independent. The statistical independence between two random vectors implies

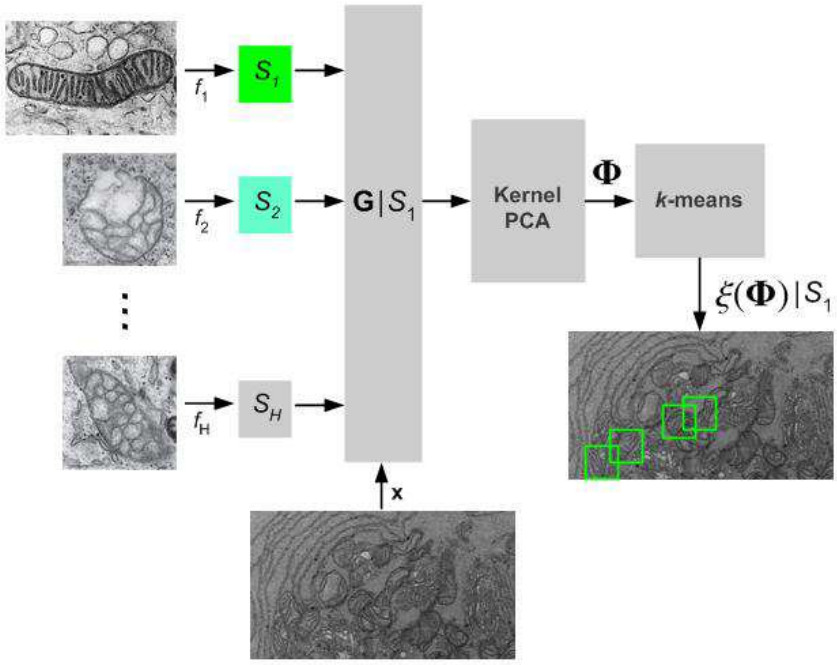


Figure 3.3.1: Simplified architecture of our proposed model-based spectral clustering approach to the unsupervised segmentation of mitochondria [50]. For a given dataset of size H of different mitochondrial morphologies, we derive a set of low-level features $f_1, ..., f_H$ (e.g. distance between the pixels, similarity in brightness, orientation, texture, etc.) according to the Gestalt laws of perceptual organization. Then, the features are factored into corresponding similarity models $S_1, ...S_H$. For a given test image, i.e. S_1 we construct the corresponding Gram matrix $\mathbf{G}|S_1$ and run the kPCA algorithm in order to obtain the set of feature vectors $\mathbf{\Phi}$. By applying k-means in the feature space we obtain the desired partitioning of the input (image) space.

that the mutual information between these two random vectors is zero. In other words, the independence of signals implies the maximization of information and it is reasonable to assume that this information is also related to the feature space cluster formations.

The theoretical research in very large scale implementations of spectral clustering is motivated also by other application domains. An efficient very large scale implementation of spectral clustering is not only of interest for pixel or voxel classification tasks, but for any application concerning knowledge discovery in very large datasets. Some applications could, for instance, involve large scale analysis of gene expression, large clinical and administration, or sensor datasets.

Inferences made through an efficient very large scale spectral clustering implementation may also help to understand cognitive processes and neural processing. Very large scale spectral clustering implementations based on similarity models with perceptual constraints have also interdisciplinary links to psychology, human vision and neural processing. Nevertheless, the research on applications of spectral clustering to segmentation is only about 30 years old, considering the paper by Perona and Freeman as seminal [166]. Research towards alternatives to the conventional component analysis based on spectral decomposition is also an active field which indicates that there is a demand for more efficient implementation algorithms and novel theoretical formulations. A comprehensive survey on classical and more recent dimensionality reduction methods is given in [101] and more recently in [230] and [140].

As described in Section 2.2.6.4, kernel-PCA attempts to construct a set of orthogonal and uncorrelated components. In some applications, the statistical independence of components can be directly related to the amount of information conveyed in the feature space. Therefore, one possible direction of research would be to investigate methods which consider the same objective. One of such a candidate for this task is ICA which is reviewed in the next section.

3.3.1 Independent Component Analysis

In the standard PCA formulation it is not necessary to assume any particular distribution for the input data x_i [101]. One way of handling possible non-normality in PCA, especially if the distribution has heavy tails, is to use robust estimation of the covariance or correlation matrix or of the principle components themselves. ICA, on the other hand assumes the non-Gaussianity of the data. ICA is a relatively new signal processing and data analysis technique. It has primarily found applications in wireless communication, biomedical signal processing, data mining and pattern classification. ICA may be also used for the Blind Source Separation problem.

The derivations in ICA are based on traditional probability theory and include notations for random variables. Here, all random objects are written in typewriter font, e.g. $\mathtt{x}$, in order to distinguish them from deterministic ones, e.g. x. For random vectors, e.g. $\vec{\mathtt{x}}$, we use the vector symbol in order to separate them from scalar random variables. For deterministic objects, bold face lower case letters are used for vectors and the bold face upper case letters are used for matrices. Using these notations, the ICA problem can then be defined by using a statistical latent variables model. Assume that we observe n linear mixtures $\mathtt{x}_1, ..., \mathtt{x}_n$ of independent components:

$$\mathtt{x}_j = a_{j1}\mathtt{y}_1 + a_{j2}\mathtt{y}_2 + ... + a_{jn}\mathtt{y}_n, \forall j \tag{3.15}$$

We assume that each mixture $\mathtt{x}_j$ as well as each independent component $\mathtt{y}_k$ is a random variable. The observed values $\mathtt{x}_j$ are then a sample of this random variable. It is assumed that the mixture variables and the independent components have zero means. If this is not true then the observable variables $\mathtt{x}_i$ can always be centered by subtracting the sample mean.

Let $\vec{\mathtt{x}}$ denote the random vector whose elements are the mixtures $\mathtt{x}_1, ..., \mathtt{x}_n$, and likewise let $\vec{\mathtt{y}}$ be the random vector with elements $\mathtt{y}_1, ..., \mathtt{y}_n$. Let $\mathbf{M}$ denote the mixing matrix with elements a_{ij}. Using this vector-matrix notation the above noiseless mixture model is written as:

$$\vec{\mathtt{x}} = \mathbf{M}\vec{\mathtt{y}} \tag{3.16}$$

The goal of ICA is to recover the original source vectors $\vec{\mathtt{y}}$ from the observation vectors $\vec{\mathtt{x}}$ blindly without explicit knowledge of the sources of the linear mixing system $\mathbf{M}$. The ICA model is a generative model, which means that it describes how the observed data are generated by a process of mixing the components y_j. Also the mixing matrix $\mathbf{M}$ is assumed to be unknown. All we observe is the random vector $\vec{\mathtt{x}}$, and we must estimate both $\mathbf{M}$ and $\vec{\mathtt{y}}$ using it. After estimating the matrix $\mathbf{M}$, we can compute its inverse, say $\mathbf{W}$, and obtain the independent components simply by:

$$\vec{\mathtt{y}} = \mathbf{W}\vec{\mathtt{x}} \tag{3.17}$$

The idea that physically meaningful signals underlying a data set should be independent is a major motivation for ICA.

The degree of independence is measured by the mutual information between the components of $\vec{y}$:

$$\mathrm{MI}(\vec{y}) = \int p(\vec{y}) log \frac{p(\vec{y})}{\prod_k p_k(y_k)} d\vec{y}. \tag{3.18}$$

When the joint probability $p(\vec{y})$ can be factored into the product of the marginal densities $p_k(a_k)$, the various components of $\vec{y}$ are statistically independent and the mutual information is zero.

Bell and Sejnowski in [17] considered a nonlinear, component-wise mapping $\vec{y} = \mathbf{g}(\vec{s})$, $y_k = g_k(s_k)$ into a space in which the marginal densities are uniform. The linear transformation followed by the nonlinear map may be accomplished by a single-layer neural network in which the elements of $\mathbf{W}$ are the weights and the K neurons have transfer functions g_k. Since the mutual information is constant under invertible, component-wise change of variables, $\mathrm{MI}(\mathbf{s}) = \mathrm{MI}(\vec{y})$, and since the g_k are, in theory at least, chosen to generate uniform marginal densities, $p_k(y_k)$, the mutual information $\mathrm{MI}(\vec{y})$ is equal to the negative of the entropy (negentropy) of $\vec{y}$:

$$\mathrm{MI}(\vec{y}) = -H(\vec{y}) = \int p(\vec{y}) log p(\vec{y}) d\vec{y} \tag{3.19}$$

Principles of ICA estimation include various optimization criteria such as:

1. Minimization of the mutual information given by (3.18)
2. Maximization of the negentropy given by (3.19)
3. Minimization of the higher order moments, which measure the non-Gaussianity
4. Maximum likelihood estimation, where the log-likelihood for a single observation $\mathbf{x}(t)$ is:

$$log P(\mathbf{x}(t)|W) = log|detW| + \sum_k log p_k(a_k(t)) \tag{3.20}$$

The above list represents the so called *contrast* functions. For two random variables x_1 and x_2, a valid contrast function is always non-negative, and equal to zero iff the variables x_1 and x_2 are independent [11]. The choice of L dominant independent components $\vec{s}_L = W_L \vec{x}$ is an important subject. Tian et al. [206], e.g., adopted the procedure of calculating negentropy of each component and arranging the components in descending order according to their negentropy values. The authors further set a non-Gaussianity threshold and only kept the components whose negentropy values were larger than the threshold. The desired threshold value could only be found empirically.

There is an important connection between statistical independence, physical meaningfulness of signals and maximum of information carried by those signals. The notion of information maximization is related to the joint entropy $H(\vec{x_1}, \vec{x_2})$ between two random vectors $\vec{x_1}$ and $\vec{x_2}$:

$$H(\vec{x_1}, \vec{x_2}) = H(\vec{x_1}) + H(\vec{x_2}) - \mathrm{MI}(\vec{x_1}, \vec{x_2}) \tag{3.21}$$

Some ICA algorithms require a preliminary sphering or whitening of the data. A random vector $\vec{x}$ is said to be spatially white, if $\mathrm{cov}[\vec{x}] = \sigma^2 \mathbf{I}_n$. Here, $\mathbf{I}_n$ denotes the $n \times n$ identity matrix and $\sigma^2 > 0$. Sphering means that the observed variable $\vec{x}$ is linearly transformed to a variable $\vec{v}$ as $\vec{v} = \mathbf{Q}\vec{x}$ such that the covariance matrix of $\vec{v}$ equals unity $\mathrm{E}[\vec{v}\vec{v}^T] = \mathbf{I}$. Therefore the terms *sphering* and *whitening* have essentially the same meaning. The result of sphering is the whitened space. The sphering can be accomplished by classical PCA. Usually, by whitening data the mutual information is reduced, so that the whitened data are closer to independence [2].

The objective of PCA is to maximize the variance and to remove the unreliable dimensions caused by insufficient or unrepresentative data [99]. The orthogonality and uncorrelatedness constraints are included to ensure that different components are measuring different things. By contrast, ICA takes the separation of components as its main aim. ICA starts from the view that uncorrelatedness is rather limited as it only considers a lack of linear relationship, and that ideally components should be statistically independent. This is a stronger requirement than uncorrelatedness, with the two only equivalent for Gaussian random variables. ICA thus can be viewed as the higher-order generalization of PCA to non-normal data.

3.3.2 Combined Approaches

Alzate and Suykens [7], for example, proposed a novel multiway spectral clustering implementation with out-of-sample extensions without relying on low-rank approximation such as Nyström method. Their model is based on a weighted kernel-PCA scheme. The formulation was initially based on a binary classification scheme and further extended to the multiway clustering case with encoding and decoding schemes. The formulation fits into the least-squares SVM framework by considering weighted versions. The proposed approach was cast in a constrained optimization framework. This interpretation allowed the clustering model to be extended to out-of-sample points. The eigenvectors of a modified similarity matrix derived from the data were shown to be a dual solution to a primal optimization problem formulated in a high-dimensional feature space. These solutions contain clustering information and show a special structure when the clusters are well-formed. A number of combined dimensionality reduction algorithms are listed in Table 3.3.1.

Table 3.3.1: Combined Dimensionality Reduction Algorithms.

Algorithms	Concept	Manifold
LDA + CDA [99]	single discriminant evaluation	assymetric
CCA (Canonical Correlation Analysis) [230]	optimal coordinate system	linear
KICA (Kernel-ICA) [11]	mutual information	non-linear
Kernel-PCA + ICA [228]	ICA in the feature space	non-linear
Multiway KICA [206]	feature subsampling	non-linear

The work by Jiang [99] analyzed the role of PCA in classification and addressed the problem of applying PCA on the asymmetric classes or the unbalanced training data. The author noted that in order to extract a set of features quickly and efficiently there is a need for a discriminant analysis which maximizes the discriminatory power of the extracted features. In addition, the work proposed an asymmetric discriminant analysis method that integrates LDA (Linear Discriminant Analysis) and CDA (Covariance Discriminant Analysis) in a single discriminant evaluation and regularizes the two covariances matrices.

Dhillon et al. [49] proposed a new formulation of the Graph-Cut algorithm. This method does not use eigenvectors, but assumes the availability of the similarity matrix. This approach is related to, but different, from the standard spectral clustering. Without the eigendecomposition step, it does not produce a low-dimensional representation of the data. In other words, their method does not conduct dimensionality reduction as spectral clustering does.

CCA (Canonical Correlation Analysis) is a multivariate statistical technique similar in spirit to PCA. While PCA works with a single random vector and maximizes the variance of projections of the data, CCA works with a pair of random vectors or in general with a set of m random vectors and maximizes correlation between sets of projections. While PCA leads to an eigenvector problem, CCA leads to a generalized eigenvector problem [11, 101, 230].

A method which merges the ideas from kernel-PCA and ICA and utilizes the idea from CCA is given in the paper by Bach and Jordan [11]. The authors noted that the fundamental problem with ICA is that it generally fails to separate the nonlinearly mixed source due to its intrinsic linearity. The authors presented an algorithm called KICA (Kernel Independent Component Analysis) which is not the kernelization of the ICA algorithm. Rather, it is a new approach to ICA, based on novel kernel-based measures of dependence. Specifically, the authors addressed the problem of implementing the mutual information as the contrast function. The mutual information for real-valued variables is difficult to approximate and optimize on the basis of finite samples. The authors proposed two novel contrast functions. Minimizing them leads to two different KICA algorithms. The authors defined the $\mathcal{F}$-correlation as a measure of statistical dependence among random variables $x_1, ..., x_m$. Given a reproducing-kernel Hilbert space $\mathcal{F}$ on $\mathbb{R}$, with kernel $K(x, y)$ and feature map $\Phi(x)$, the $\mathcal{F}$-correlation is defined as the maximal correlation between the random variables $f_1(x_1)$ and $f_2(x_2)$, where f_1 and f_2 range over $\mathcal{F}$:

$$\rho_{\mathcal{F}} = \max_{f_1, f_2 \in \mathcal{F}} [corr(f_1(x_1), f_2(x_2))] \, . \tag{3.22}$$

The KICA algorithm can be summarized as follows: Given a set of data vectors $\mathbf{y}^1, \mathbf{y}^2, ..., \mathbf{y}^N$, and given a parameter matrix $\mathbf{W}$, set $\mathbf{x}^i = \mathbf{W}\mathbf{y}^i$, for each i, and thereby form a set of estimated source vectors $\{\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^N\}$. The m components of these vectors yield a set of m centered Gram matrices, $\mathbf{G}_1, \mathbf{G}_2, ..., \mathbf{G}_m$. These Gram matrices (which depend on $\mathbf{W}$) define the contrast function, $C(\mathbf{W}) = \hat{I}_{\rho\mathcal{F}}(\mathbf{G}_1, \mathbf{G}_2, ..., \mathbf{G}_m)$, as the solution to a generalized eigenvalue problem, $\mathbf{K}\alpha = \lambda\mathbf{D}\alpha$, where $\mathbf{K}$ and $\mathbf{D}$ are block matrices constructed from the Gram matrices $\mathbf{G}_i$. The KICA (kernel-CCA) algorithm involves minimizing this function $C(\mathbf{W})$ with respect to $\mathbf{W}$. The main idea of this algorithm is to maximize independence by minimizing correlation with the kernel.

The applications of the Bach and Jordan's realization of KICA to the computer vision domain are given as a face recognition study [142] and removal of reflections from images [226]. Martiriggiano et al. [142] were first to apply KICA on bi-dimensional signals such as images. Yamasaki et al. [226] concluded that KICA is more effective than ICA in removing reflections from the images even if the observed image is non-linearly transformed through the lenses of camera.

Yang et al. [228] presented an alternative formulation of KICA as a two stage algorithm: whitened kernel-PCA plus ICA. Their algorithm is formulated in the kernel-inducing feature space – that is they performed ICA in the feature space. Firstly, kernel-PCA spheres data and makes the data structure become as linearly separable as possible by virtue of an implicit nonlinear mapping determined by a kernel. Then, ICA seeks the projection directions in the kernel-PCA whitened space, making the distribution of the projected data as a non-Gaussian as possible.

The multiway implementation of this KICA formulation was presented by Tian et al. [206] to detect faults in the batch-fed penicillin fermentation process. In order to solve the computational problem associated with a large kernel matrix, the authors introduced a novel feature sample

extracting technique before implementing the kernel transformation.

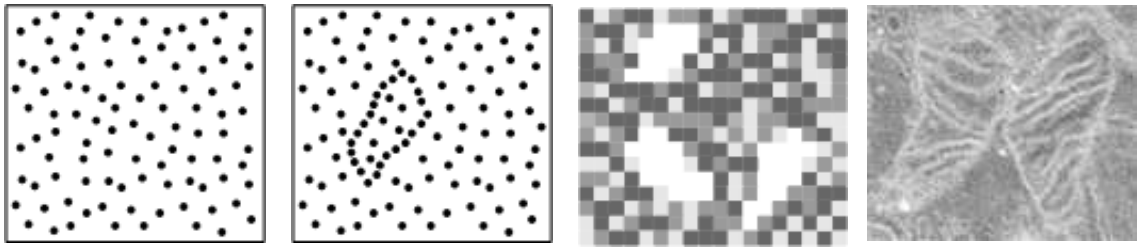
The very large scale dimensionality reduction methods reviewed are based on Nyström, column-sampling and low-rank approximation techniques. We have also discussed the alternative and combined approaches such as ICA, kernel-ICA, CCA, and a combination of kernel-PCA and ICA. These approaches are relevant to our research on the *anticorrelation*-based dimensionality reduction which is developed in Chapter 6. However, this thesis is primarily cast within the framework of spectral clustering, and the input to every spectral clustering algorithm is a similarity matrix. In many cases, similarity relationships can be derived from the visual appearance of cellular and sub-cellular organisms in electron microscopy images. In the next section, we discuss the applicability of perceptual organization to biomedical imaging and review mathematical models which integrate various perceptual grouping constraints into similarity matrices.

3.4 Perceptual Grouping Strategies in Biomedical Imaging

The study of perception is concerned with identifying the process through which we interpret and organize sensory information to produce our conscious experience of objects and object relationships. Perceptual grouping plays a critical role in both human and computer vision. Research in perceptual grouping began in the 1920's with Gestalt psychologists, whose goal was to discover the underlying principles that would unify the various grouping phenomena of human perception.

Historically, the visual phenomenon associated with perceptual organization is grouping. Indeed, perceptual grouping and perceptual organization are sometimes presented as though they were synonymous, though this is incorrect as noted by Wagemans et al. [217]. Grouping is only one particular kind of organizational phenomenon and the figure-ground organization. In general, grouping determines what the qualitative elements of perception are, and figure-ground determines the interpretation of those elements in terms of their shapes and relative locations in the layout of surfaces in the 3D world. Presently, biological vision is the only measure of the incompleteness of the current stage of computer vision, and illustrates that the problem is still open to solutions.

In biomedical imaging, the data annotation problem is inherently connected with human vision and the perception paradigm. The major challenge in applying computer vision methods to this imaging domain is the variety of different shapes and textures. Furthermore, variations such as different animal species, sample preparation, staining protocols etc., can lead to very different image characteristics. A potential solution to the above complexity challenge can be cast within the machine learning framework. Depending on the type of biological structures, a set of different features can be derived from images. These descriptors could target, for example, the segmentation of membranes, cells, mitochondria or generally irregular structures. Textures, for example, can be modeled as a joint distribution of filter responses, where the distribution is represented as a frequency histogram of filter responses. Other features such as intensity, distance, contours, boundary information, edges, color and gradient flux can be factored into the affinity models. The parameters for these models can be learnt from training images, or adjusted manually according to the available experimental data. There are a number of features which can be connected to perceptual organization. Affinity models which integrate such perceptual grouping constraints, generally follow the Gestaltic rules of pre-attentive perception.



(a) (b) (c) (d)

Figure 3.4.1: Selected laws of perceptual organisation. (a) No significant salient relations present. (b) Perception of a salient closed contour due to *Proximity*, *Closure* and *Continuation* principles. (c) Perception of three bright objects overrides the proximity of dark pixels. This phenomenon is based upon *Figure-Ground* Gestalt law which describes the need for sufficient contrast in visual stimuli. (d) TEM image showing two adjacent mitochondria with salient membrane structures.

3.4.1 Related Work on Perceptual Grouping with Similarity Models

Gestalt theory is a psychological approach originated in Germany in the late 19th century. Its concept is that human cognition has a self-organizing tendency to consider properties of independent parts in conjunction with structural laws of the whole [221]. According to this theory, grouping is one of the key elements of the human visual system. Gestalt psychologists were among the first to address the issues of pre-attentive perception. Their theory refers to the fact that the visual stimuli received by the retinal receptors further tend to be cognitively grouped in specific ways or structures [108, 221]. The manner in which perceptual grouping occurs is governed by a number of organizational laws including *Proximity*, *Similarity*, *Continuation*, *Symmetry*, *Closure* and *Familiarity* [24]. The *Figure-Ground* law defines the mechanisms behind visual segregation [62].

The grouping processes associated with the human visual perception attempt to form structural organizations of different complexities. An example is provided by the cognitive process of joining the dots along a perimeter to infer a circular shape and also more complex scenarios such as extrapolation of three-dimensional computer generated images from a series of pixels arranged on the screen [210]. The visual stimuli in Fig. 3.4.1 demonstrate how some of organizational laws can be applied to group low-level primitives into perceptually uniform shapes. Fig. 3.4.1(a) shows a random stimulus field containing uniformly distributed dots with no Gestalt properties, and thus no salient structures can be perceived in the image. Conversely, the distribution of dots in Fig. 3.4.1(b) allows us to clearly locate a closed salient contour. The underlying cognitive process behind such a phenomenon is known as *contour integration* [124] and is governed by Gestalt principles such as *Proximity*, *Continuation* and *Closure*.

The *Proximity* rule states that items placed near to each other tend to be grouped together. The law of *Continuation* describes the perceptual tendency to follow a direction derived from a visual field. The law of *Closure* describes how human cognition completes the gaps in the collection of closely located points. Lowe's work [131] was the first to introduce computational models of perceptual grouping processes, derived from image statistics.

The simplest pairwise similarity model S_{ij} which is based only on the *Proximity* principle for grouping datapoints x_i and x_j in an experimental setting similar to the one in Fig. 3.4.1(b) is, for

example, given by Perona and Freeman [166] as well as by Ng et al. [155] as:

$$S_{ij} = \exp\left[\frac{-\|x_i - x_j\|^2}{d_0^2}\right] \tag{3.23}$$

where d_0 is a reference distance below which two points are thought to be similar and beyond which two points are thought to be dissimilar. Another example of perceptual grouping is provided in Fig. 3.4.1(c) where it is illustrated how in a complex visual arrangement containing black, gray and white pixels our attention is easily drawn to brighter areas. This phenomenon is mainly based on the law of *Figure-Ground* segregation which is connected to the notion of the sufficient *global contrast* in the visual stimuli in order for Gestalt factors to act on it [62]. Gestalt laws can be extended to the perceptual interpretation of biomedical images, and for instance Fig. 3.4.1(d) shows an inverted TEM image containing two adjacent mitochondria with salient membrane structures.

Over the past few decades, the grouping principles of the Gestalt theory found interesting practical applications, particularly in the field of visual neuroscience [222], visual screen design [32] and computer vision [143]. From the implementation point of view, the most frequently used tools for salient classification in gray scale images are the generation of saliency maps [185] in the image domain and various implementations of spectral clustering [118, 166, 190] in the feature domain.

3.4.1.1 Intensity and Proximity Models

The authors in [166] explored the behavior of their spectral clustering algorithm on synthetic brightness images with the aim of finding the optimal foreground-background segmentation. For that purpose the following similarity model was defined:

$$S_{PF}(p_1, p_2) = \exp\left[-\frac{d^2(p_1, p_2)}{d_0^2} + \frac{(I(p_1) - I(p_2))^2}{dI_0}\right] \tag{3.24}$$

where i and j are image pixel indices, I is indicating brightness values and $d(p_1, p_2)$ is the geometric distance between two pixels p_1 and p_2. Parameter values of $d_0 = 3$ and $dI_0 = 1$ have been used by the authors in the experimental section. We note that the above model uses the Gestaltic grouping principles of *Proximity* (distance between pixels) and *Similarity* (brightness values). S_{PF} had also been used by Shi and Malik [190]. The authors in [166, 190], however, did not mention whether S_{PF} was defined for raw or normalized data and instead adjusted the model parameters manually.

The other similarity model, which we use here for comparison, is called a segmentation energy term by Boykov et al. [25] and Kaynig et al. [103] (see also (3.27)) and has the form:

$$S_{BFL}(p_1, p_2) = \exp\left[-\frac{(I(p_1) - I(p_2))^2}{2\sigma^2}\right] \cdot \frac{1}{d(p_1, p_2)} \,. \tag{3.25}$$

The only difference of S_{BFL} to S_{PF} is that the denominator term is not exponentially mapped. Both S_{PF} and S_{BFL} penalize discontinuities in the segmentation of neighbored pixels of similar intensities. By comparing (3.27) and (3.25) we can see that the segmentation energy term E_{gc} from [103] is the extension of S_{BFL} by the gradient flux term $|\langle v_p, u_{pq}\rangle|$. The authors noted that for the segmentation of thin and elongated structures, like membranes, it is common to incorporate the flux of the gradient vector field into the segmentation.

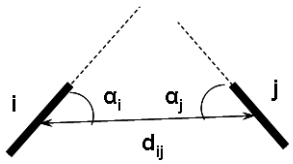


Figure 3.4.2: Variables involved in the computation of the similarity S_{ij} of two line segments i and j according to Perona and Freeman [166] applied to the grouping if co-linear line segments.

3.4.1.2 Grouping Co-linear Line Segments

In modern work, the Gestaltic principle of *Continuation* has been largely linked with work on contour integration and visual interpolation [27]. Contour integration studies examine what factors promote grouping of separate (not connected) oriented elements into contours which are detectable in a field of otherwise randomly oriented elements. In line and contour grouping, collinearity, co-circularity, smoothness, and a few other features play prominent roles in models of good continuation effects on contour integration. An intuitive similarity model S_{ij} for grouping co-linear line segments (see also Fig. 3.4.2) is provided by Perona and Freeman [166] as:

$$S_{ij} = \exp\Big[-\frac{d_{ij}^2}{d_0^2} - \frac{2 - \cos(2\alpha_i) - \cos(2\alpha_j)}{1 - \cos(2\theta_0)} - \frac{1 - \cos(2\alpha_i - 2\alpha_j)}{1 - \cos(2\delta\theta_0)} \Big] \tag{3.26}$$

This model assumes that two line segments are similar when they are close by, when they are aligned and, failing that, when they are co-circular, i.e. tangent to the same circle. In (3.26), the first term in the exponential is a distance-related affinity, the second term penalizes the average deviation of the line segments from being collinear, and the third term penalizes the non-co-circularity of the two line segments. Perona and Freeman [166] suggested that 'good' values for d_0 should range between the spacing of the elements and five times that value. 'Good' values for θ_0 typically range from $\pi/2$ to $\pi/10$, while $\delta\theta_0$ typically should be half to one-fourth as large as θ_0.

3.4.1.3 Continuation and Membrane Gap Completion

Kaynig et al. [103] approached the problem of perceptual grouping with the analysis of fundamental Gestalt properties of data studied, and embed *Ncut* within a machine-learning framework in order to extract membranes in serial section TEM images. In the graph cut framework each pixel p is mapped to corresponding labels $y_p \in \{0, 1\}$ such that the entire labeling y for all pixels minimizes a given energy function $E(y)$. In their work, the authors focused on the development of a new energy term E_{gc} that models the *Continuation* principle with the aim of obtaining more accurate gap completion in the membranes imaged. Intuitively, lines as well as membranes are directed structures. By the principle of good *Continuation*, well-classified parts of directed structures should enforce smoothness in labels along their orientation. This can be formulated by $E_{gc}(y_p, y_q)$:

$$E_{gc}(y_p, y_q) = |\langle v_p, u_{pq} \rangle| \cdot \exp\big[-\frac{(x_p - x_m)^2}{2\sigma^2} \big] \cdot \frac{\delta_{\rightarrow}(y_p, y_q)}{d(p, q)} \tag{3.27}$$

where u_{pq} is a unit vector with the orientation of a straight line between pixels p and q, and v_p is a vector directed along the membrane. The length of v_p reflects the orientedness of the image at p. For this purpose the authors used a directed filter consisting of a straight line with a thickness

equal to the average membrane thickness in the training images. The inner product $\langle v_p, u_{pq} \rangle$ is then estimated by the response to this filter oriented according to u_{pq}. The value of x_m is given as the average gray value of membrane pixels, and σ^2 can be estimated as the variance of these gray values. Thus, the difference $(x_p - x_m)$ weights the energy term according to the similarity of x_p to the typical gray value of a membrane. The factor $\delta_{\rightarrow}(y_p, y_q)$ is not symmetric and $\delta_{\rightarrow}(y_p, y_q) = 1$ for $y_p = 1, y_q = 0$ and $\delta_{\rightarrow}(y_p, y_q) = 0$ for all other cases. This asymmetric definition ensures that E_{gc} only penalizes cuts that violate smoothness along the direction of membrane pixels. The proposed energy terms allow for contour completion in situations where gradient flux-based methods fail.

3.4.1.4 Combining Color and Boundary Information

By using the color cue where close-by pixels with similar colors are likely to belong to the same image region, one can define the color-based model S_{ij}^{C} [105]:

$$S_{ij}^{C} = \exp\left[-\theta_x \|x_i - x_j\|^2 - \theta_g \|g_i - g_j\|^2 \right] \tag{3.28}$$

where x_i and g_i denote position and color of a pixel i respectively. Kim et al. [105] noted that connecting pixels by color is useful when linking disjointed object parts, but results in errors if the background has a similar color distribution to the object parts. The authors also noted that edgeness is one important boundary cue to detect a potential object boundary. The boundary-based similarity model S_{ij}^{B} is formulated by measuring the magnitude of image edges between two pixels:

$$S_{ij}^{B} = \exp\left[-\max_{i' \in \overline{ij}} \theta_f \|f_{i'}\|^2 \right] \tag{3.29}$$

where $\overline{ij}$ is a straight line joining two pixels i and j, and f_i is the edge strength of a pixel i. This boundary-based model is particularly useful when background clutter has a similar color to the object body. However, since it uses only edgeness along the straight line between two pixels without considering all possible paths, texture edges often disturb the long-range affinity estimation [105]. To make the appropriate similarity model in all natural images, it is helpful to combine these two grouping cues with a parameter α for the combined similarity model S_{ij}^{M} as follows:

$$S_{ij}^{M} = \sqrt{S_{ij}^{C} \times S_{ij}^{B}} + \alpha S_{ij}^{B} \,. \tag{3.30}$$

However, this model still has some weakness in the long-range affinity estimation, since it is formulated by naively mixing two simple color and boundary affinity models.

3.4.1.5 Grouping Superpixels

For the purpose of grouping superpixels[1] into a saliency structure that approximates the mitochondria contours, Ghita et al. [76] defined the following three separate similarity functions:

$$S_1(i, j) = \exp\left[-\frac{I(i) + I(j)}{2a_1^2} \right] \tag{3.31}$$

$$S_2(i, j) = \exp\left[-\frac{|I(i) - I(j)|}{2a_2^2} \right] \tag{3.32}$$

[1] Please, refer to page 7 for the definition of a superpixel.

$$S_3(i,j) = \exp\left[-\frac{d(c_i, c_j)}{2a_3^2}\right] \tag{3.33}$$

where $I(i)$ and c_i are the mean intensity and the centroid of the superpixel with index i. These three functions return values in the interval $(0, 1]$ and they implement low-level properties that encode the intensity and spatial constraints that are characteristic of mitochondria contours. The function S_1 implements the constraint that the superpixels associated with mitochondria contours are defined by low intensity values. The function S_2 enforces an intensity continuity constraint (i.e. S_2 returns values closer to 1.0 when the mean intensity values of the superpixels i and j are similar). The function S_3 implements a spatial continuity constraint that assigns larger values for superpixels whose distance between their centroids c_i and c_j is small. The parameters a_1, a_2 and a_3 are parameters that weigh the strength of each constraint in the calculation of the similarity matrix. In addition, the intensity values and superpixel distances are normalized in the range $[0, 1]$. The combined final similarity model S_{ij} is then calculated as follows:

$$S_{ij} = S_1(i,j) \times S_2(i,j) \times S_3(i,j) \tag{3.34}$$

3.4.1.6 Integrating Texture and Contours

Malik et al. [137] proposed a similarity model that integrates texture and contour information. Pairwise texture similarities were computed by comparing windowed texton histograms, where the windows are centered around the two pixels being compared. Then, one approach to cue integration is to define the weight between pixels i and j as the product of the contribution from texture and that from the contour:

$$S_{ij} = S_{ij}^{IC} \times S_{ij}^{TX} \tag{3.35}$$

The formulation of the orientation energy S_{ij}^{IC} in (3.36) allows one to extract sub-pixel localization of the contour by finding the local maxima in the orientation energy $OE(x)$ perpendicular to the contour orientation, where the confidence of this contour is given by:

$$S_{ij}^{IC} = \exp\left[-\max_{x \in M_{ij}} \frac{OE(x)}{\sigma_{IC}}\right] \tag{3.36}$$

and where M_{ij} is the set of local maxima along the line joining pixels i and j. In other words, two pixels have a weak link between them if there is a strong local maximum of orientation energy along the line joining the two pixels. On the other hand, if there is little energy, for example in a constant brightness region, the link between the two pixels is strong. In order to derive texture-based similarity model, authors use the χ^2 test for comparing windowed texton histograms:

$$\chi^2(h_i, h_j) = \frac{1}{2}\sum_{k=1}^{K} \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_J(k)} \tag{3.37}$$

where h_i and h_j are the two histograms. Then, S_{ij}^{TX} is defined using χ^2 distance between texton histograms at pixels i and j:

$$S_{ij}^{TX} = \exp\left[-\frac{\chi^2(h_i, h_j)}{\sigma_{TX}}\right] \tag{3.38}$$

Table 3.4.1: Summary of Reviewed Similarity Models.

Ref.	Eq.	Proximity	Intensity	Texture	Contin	Lines	Color	Edges	Contour	SP
[155]	(3.23)	√								
[166]	(3.24)	√	√							
[25]	(3.25)	√	√							
[166]	(3.26)	√				√				
[103]	(3.27)	√	√		√					
[105]	(3.30)	√					√	√		
[76]	(3.34)	√	√							√
[137]	(3.35)	√		√					√	

The abbreviations used are SP (for Superpixels) and Contin (for Continuation).

In graph-based spectral clustering approaches, the overall segmentation quality depends mainly on the graph affinities [105]. Therefore, to produce high-quality segmentation with object-level details, it is important to integrate different local grouping cues. Table 3.4.1 shows that all models use the Proximity cue to define distances between image pixels, lines, data points or superpixels.

3.4.1.7 Learning Affinity Models

The trend in machine learning applications to biomedical image segmentation is in learning affinity functions. Fowlkes et al. [69], for example, studied the problem of combining region and boundary cues for natural image segmentation within the semi-supervised machine learning framework. The authors employed a large database of manually segmented images in order to *learn* an optimal affinity function between pairs of pixels. These pairwise affinities can then be used to cluster the pixels into visually coherent groups. Region cues are computed as the similarity in brightness, color and texture between image patches. Boundary cues are incorporated by looking for the presence of an intervening contour, a large gradient along a straight line connecting two pixels.

The authors in [69] used a dataset of human segmentations to individually optimize parameters of the patch and gradient features for brightness, color, and texture cues. Then they quantitatively measured the power of different feature combinations by computing the precision and recall of classifiers trained using those features. The mutual information between the output of the classifiers and the same-segment indicator function provided an alternative evaluation technique that yielded identical conclusions. The best classifier made use of brightness, color, and texture features, in both patch and gradient forms. For brightness, the gradient cue outperformed the patch similarity. In contrast, using color patch similarity yielded better results than using color gradients. Texture was the most powerful of the three channels, with both patches and gradients carrying significant independent information. Interestingly, as reported by authors, the proximity of two pixels did not add any information beyond that provided by the similarity cues.

Another work on *learning* affinity functions is given by Kim et al. [105] where the authors employed a semi-supervised learning technique in order to learn optimal affinities from the test image without iteration. Since these well-defined full-pairwise affinities are directly used in the multi-layer spectral segmentation framework, the algorithm produced high-quality segmentation results with object details in natural images.

3.5 Summary of Background Research

We have reviewed the state of the art on cellular and mitochondria segmentation in Section 1.3 and Section 2.1. A summary of the reviewed methods is given in chronological order in Table 3.5.1.

Table 3.5.1: Summary of background research on cellular and subcellular segmentation methods.

Authors	Structures	Supervised	Unsupervised	Other Methods	Year
[157]	cells	CNN		EBM	2005
[216]	**mito**	Gentle-Boost		Gabor filters	2008
[156]	memb, **mito**, cells			Watersnake	2008
[4]	cells	NN	PCA		2009
[193]	cells, **mito**	Adaboost		Ray	2009
[153]	cells, **mito**	Nearest Neighbor SVM, AdaBoost		Textons Level Sets	2009
[205]	cells		Graph Cut		2009
[103]	membranes	Random Forest	Graph Cut	Haar, PG	2010
[113]	memb, **mito**			Radon	2010
[3]	cells		Graph Cut		2010
[141]	membranes			Scale-Space	2011
[111]	membranes	Random Forest		Superpixels	2011
[132]	**mito**, cells	SVM	*k*-means Graph Cuts	Superpixels Ray	2012
[37]	membranes	DNN			2012
[14]	synapses	AdaBoost		Superpixels	2012
[122]	membranes	CNN	Graph Cut	PG	2012
[92]	cells		*k*-means		2013
[184]	**mito**	Random Forest		Algebraic Curves	2013
[162]	particles			Morphology	2013
[154]	synapses	SVM, AdaBoost		Texture,Shape	2013
[76]	**mito**		Affinity Factorization	Superpixels PG	2014
[203]	cells	Random Forest		ilastik	2014
[227]	cells			Watershed	2014
[229]	cells	Random Forest	Multicut	Superpixels	2014
[172]	cells, membranes	CNN			2015
[198]	membranes		RNN		2015
[188]	cells	SVM Baum-Welch		HMM, Viterbi Superpixels	2015
[109]	cells	CNN		MIL	2016
[219]	cells			Thresholds	2016
[196]	particles			Watershed	2016
[125]	**mito**	AdaBoost		Sparse Region Growing	2016
[138]	synapses, **mito**	Bayes	PCA, Graph Cut	CRF	2016

The columns in gray show the combined (supervised and unsupervised) machine learning methods. Abbreviations used are *memb* (membranes) and *mito* (mitochondria).

In this thesis, we have reviewed many more neural networks architectures, dimensionality reduction algorithms, cellular and subcellular segmentation approaches, graph-based spectral clustering algorithms, classifiers, affinity models and feature extraction techniques. Table 3.5.1 shows only the 33 reviewed methods which are related to the cellular and subcellular image segmentation. It can be seen that many modern segmentation approaches often combine unsupervised and supervised machine learning formulations in addition to the feature extraction techniques.

Table 3.5.2: Statistics of the reviewed segmentation methods.

Methods		References
Spectral Clustering	9	[205, 103, 3, 132, 133, 122, 76, 229, 138]
Combined Supervised and Unsupervised	7	[4, 103, 132, 133, 122, 229, 138]
Superpixels	7	[111, 133, 132, 14, 76, 229, 188]
Deep Learning (CNN, DNN)	5	[157, 37, 122, 172, 109]
AdaBoost	5	[193, 153, 14, 154, 125]
Random Forest	5	[103, 111, 184, 203, 229]
SVM	5	[153, 154, 188, 132, 133]
Not Machine Learning-based Methods	7	[156, 113, 141, 162, 227, 219, 196]

The numbers in Table 3.5.2 show how many authors out of the 33 reviewed papers are falling into the corresponding categories of methods. In general, the methods reviewed targeted segmentation of three types of structures. These are cells (nanoparticles), mitochondria and membranes (synapses). It can be seen that there are many diverse solutions which targeted specific structures and which are based on different concepts. Most approaches to the segmentation of mitochondria have applied supervised machine learning with such classifiers as Adaboost, SVM and Random Forest. To compare the performance of the methods reviewed is difficult because they are based on different architectural ideas and are validated on different datasets.

The synapses and membranes are elongated thin structures and are usually darker than intracellular space. The features reviewed included Continuation Energy, Gradient flux, Haar, Radon, intensity and superpixels. These features can also be applied to mitochondria in cases where the outer boundary is clearly defined by the membrane. For irregularly shaped mitochondria, Ray features have been shown to outperform the Haar features [193]. Superpixels were frequently used in combination with spectral graph-based methods to reduce the inial complexity of the input data. Superpixels are also used in the open source machine learning and segmentation software toolkit *ilastik* which is available since 2011. This user-interactive tool also uses Random Forest-based training on the labeled data, and can be used for the detection of cells [203] and membranes [111].

All supervised and combined machine learning methods rely on large manually annotated data. Our review shows that unsupervised machine learning can be superior in discovering the underlying structure in the input data without prior knowledge about the targets. There are also some reviewed papers which use purely unsupervised machine learning in the context of biomedical imaging. Graph-based spectral clustering approaches attempt to integrate various feature cues into the affinity models. Some papers considered Gestalt properties of the perceptual organization of the human visual system[2] and used additional classifiers to learn parameters of the models.

[2]Table 3.5.1 has an entry PG (Perceptual Grouping) for these works

Deep learning is a rapidly growing research area. Because of this emerging field, we have included works on CNNs. Most of the reviewed CNN-based methods are purely supervised. Leena et al. [122] combined a CNN to learn the affinity graph based on perceptual grouping constraints, with the Graph Cut algorithm. A modified CNN which has max-pooling layers instead of sub-sampling layers, was proposed by Ciresan et al. [37] to segment neuronal structures in EM images. Ning et al. [157] have combined the CNN with EBM (Energy Based Model) and Marquez-Neila et al. [138] have used CRF (Conditional Random Field) to model arising probabilistic dependency.

Due to emerging new hardware platforms, CNNs have recently shown a considerable speed-up factor in computational time. However, CNNs are still very complex and require an immense amount of annotated data. In addition, they have a very large number of parameters to train and their results are difficult to reproduce. Currently, deep learning solutions appear to outperform the established computer vision approaches. Some concerns have been expressed that CNNs may actually replace these approaches to segmentation and classification tasks and discourage the theoretical research [161]. However, we can see from Table 3.5.1 that non-machine learning-based methods are being also developed in parallel to CNNs and machine learning tools. Some works reviewed included such classical image processing methods as Thresholds, Mathematical Morphology, Watershed and Sparse Region Growing. Also, Active Contours, Level Sets, Watersnake and Algebraic Curves have been applied in a number of publications.

The quality of spectral image segmentation or input domain partitioning in general is directly related to the cue integration and mathematical formulations of similarity measures. The latter often contain application-dependent and non-linear kernel functions in order to allow the linear separability of clusters in the feature space. Because this topic is very relevant to this thesis, we have discussed the concepts of kernel-SVM, kernel-PCA and kernel-ICA algorithms in detail.

A survey on unsupervised neural networks provide insight into how dimensionality reduction and clustering algorithms can be implemented using neural architectures. The networks reviewed which are based on AE (AutoEncoder) architectures included Neural-PCA (which can perform kernel-PCA) and linear Neural-ICA. Kernel-SVM can also be implemented with a feedforward neural network consisting of two hidden layers [39]. We have found only one account of applying an RNN to the segmentation of neuronal structures in EM images in the work by Stollenga et al. [198]. Here, the authors developed a multidimensional RNN architecture which is based on the idea that three-dimensional data can be modeled as a sequence of two-dimensional planes. So far, we have not seen any biomedical imaging applications of SOM-based networks.

Our interest in very large scale spectral clustering has originated from the necessity to solve large scale pixel classification problems. We have discussed the numerical solutions to very large scale spectral clustering, its modifications, alternatives and possible solutions for asymmetric datasets. These methods can be used for any large datasets which arise in areas other than image processing. Some works presented hardware-based HPC implementations. Other methods are based on Nyström, column-sampling and low rank approximation techniques. One alternative to spectral dimensionality reduction is ICA, which enforces an additional constraint of statistical independence between the source signals. Other combined approaches were kernel-ICA, CCA, a combination of kernel-PCA and ICA and the Asymmetric Discriminant analysis method. We did not find any work which applied these algorithms to electron microscopy image segmentation.

Chapter 4

Localization of Mitochondria based on Clustering Extracted Line Segments

This chapter is concerned with the grouping of elementary line segments which are comprised of pairwise projected entities[1]. In a dynamic and densely-cluttered scene we consider a feature-driven recognition of objects with predominantly linear or quasi-linear structural elements. The motivation arises from the field of biomedical imaging such as the detection of mitochondria in a complex subcellular environment, and we specifically target the localization of mitochondria of lamellar and tubular morphology. Subsequent line extraction operations result in a set of line segments with different lengths, density and orientations. We observe that a distinct criterion to distinguish such a salient group of line segments from the background can be formulated as *Projectivity*. We introduce a new similarity measure *Projection-to-Distance Ratio* which combines the proximity and the amount of spanned orthogonal projections between two line segments. Furthermore, we perform investigations on the Euclidean properties of the proposed similarity measure. We construct the similarity matrix and show that it translates into an indefinite pseudo-covariance matrix. In order to test the similarity measure introduced we examine the applicability of nearest neighbor and k-means clustering methods for the grouping of line segments.

We organize this chapter as follows: in Section 4.4 we establish notations for elementary line segments and derive the relations for pairwise orthogonal projections. Furthermore, we introduce a combined similarity measure, construct a similarity matrix and analyze its properties. The grouping strategy follows the combined framework of the kernel-PCA and CSE (Constant Shift Embedding) which has been outlined in Chapter 2. We present experimental results for synthetic data sets in Section 4.7 and the results for application-specific TEM data in Section 4.8. We finish with discussion and concluding remarks in Sections 4.9 and 4.10.

4.1 Introduction

Eight decades ago Gestalt psychologists formulated the fundamental principles of the human visual recognition system. Our ability to perceive and identify non-accidental and significant objects in a complex environment relies on the fact that we intuitively favour the organization of

[1]The shorter version of this chapter has been published in Pattern Recognition Letters Journal (Elsevier, 2015) [51]

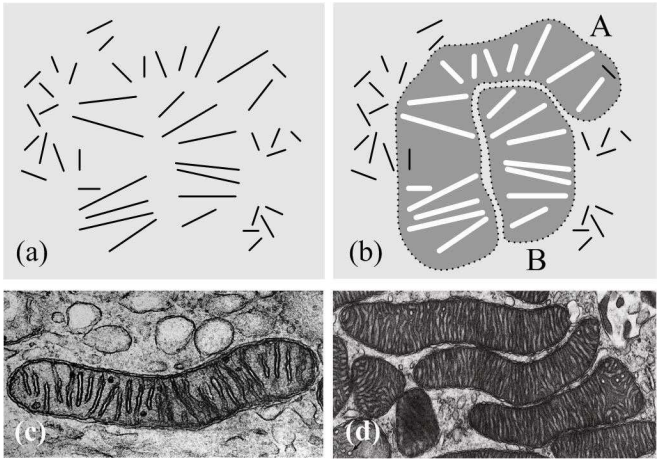


Figure 4.1.1: Line Grouping objectives. **(a)** Set of line segments with different lengths, density and orientations. **(b)** Two distinct elongated objects of interest. **(c)** Lamellar-type mitochondrion from mouse epididymis (image source: American Society for Cell Biology (ASCB)). **(d)** Tubular-type mitochondria from hamster's adrenal cortex cell (image source: courtesy of ASCB).

objects exhibiting *Proximity*, *Similarity*, *Continuation*, *Symmetry*, *Closure* and *Familiarity* properties [24, 131]. The formalism of perceptual organization governs and influences artificial grouping strategies. Grouping itself as a process can vary from grouping of low-level geometric primitives to grouping of complex objects. Any object can be parameterized and represented by a token which can be used for grouping.

A line segment, often referred to as an Elementary Line Segment (ELS), forms an important category of low-level primitives. A network of short line segments is usually extracted from an image using Hough transform, line fitting or curve smoothing operations on the edge map. Lowe [131, 130] examined the connectivity relations between line segments from the perspective of perceptual organization, and postulated inferences of *Proximity*, *Collinearity*, *Parallelism*, *Equal Spacing*, *Cotermination* and *Convergency To A Common Point* as a combination of basic Gestalt laws. The decision on how to analyze a set of linear segments and to extract salient structures from the complex background depends greatly on the underlying application. Much attention in the literature is given to the detection of salient curves and lines composed of a number of short line segments [6, 75, 185]. Sha'ashua and Ullman [185] defined a global saliency measure to identify smooth curves and use the instances of *Collinearity*, *Cotermination* and *Parallelism* among straight line segments. Jang and Hong [98] considered the detection of long line segments and indirectly applied the inferences of *Collinearity* and *Proximity*. Stahl and Wang [195] applied the *Symmetry* principle to the detection of closed convex boundaries with symmetry.

We now examine a new principle for the grouping of line segments. Fig. 4.1.1(b) shows two synthetic elongated objects A and B with flexible shapes and transverse elements. In the following we consider two perspectives from which this example can be viewed. The first is closely related to the field of perceptual organization and concerns our ability to recognize the objects A and B from the set of lines in Fig. 4.1.1(a) without any *a priori* knowledge about their shape and composition. The second belongs to the computer vision domain and attempts to establish analytic relationships between line segments which are necessary for the recovery of both objects.

Despite being conceptually different, the two approaches are more complementary than contradictory. Naturally, the visual scan of Fig. 4.1.1(a) results in a number of different interpretations depending on the observer's preference for factors such as parallelism, density, size or collinearity, and also on the observer's experience with similar structures.

At first one may argue that the lower part of Fig. 4.1.1(a) appears to be the more salient. However, it is an acceptable assumption that in a search for a smooth contour, we may at some stage discover object B by cognitively filling the gaps between its transverse elements.
This process can be described by the *Continuation* and *Closure* principles. Having realized, in other words *learned*, that this discovery may stem from a causal relationship, we may proceed with scanning the space to register object A. Encouraged by this perspective, we introduce a new similarity criterion denoted as *Projectivity*. We reason that this measure uniquely resolves the problem of forming object B, and more particularly object A, which otherwise exhibits no similarity in size, orientation or density of its line segments. Indeed, every line segment in A and B is bounded by a "just right" amount of projection of its two spatial counterparts.

The aspect of randomness is particularly interesting in this context. It has been noted by Lowe [131] that features that resemble a non-accidental object are likely to be located close in space. This fact constitutes our preference for an *orthogonal* projection which has an inherently geometric connection with the notion of a shortest distance. Although orthogonal projections are an integral part of various grouping methods [45, 195], an investigation on a stand-alone similarity which would describe orthogonally projected line segments has not been previously reported.

Lowe [131, 130] applied *parallel* projections to the detection of three-dimensional objects from two-dimensional images. Studies on trapezoid constructions indirectly anticipated the usage of orthogonal projections of line segments onto a common reference axis. Applications therein considered the problem of point location and trapezoidal maps [45] and the detection of symmetric boundaries [195]. This formalism is also related to the work by Jang and Hong [98] of measuring the overlap between two line segments in the case of co-linear grouping and endpoint linking.

Contour integration problems have been well studied in the literature. Some important works include contour processing by Heitger et al. [90], the stochastic completion fields of Williams and Jacobs [225], the work of Guy and Medioni [86] and that of August et al. [10] on contour fragment grouping. In particular, August et al. [10] explored the idea that occlusion can give rise to large gaps. This process creates a long-distance contour fragment grouping problem. The authors proposed the long-distance principle that those fragments should be grouped whose fragmentation could have arisen from a shared, simple occluder. The gap skeleton was introduced as a representation of this virtual occluder. The algorithm sought the groups that most naturally accounted for the data in the sense of perceptual occlusion. In particular, two curve endpoints could be grouped only if they had a gap skeleton computed using *projection* for all skeletal points.

Guy and Medioni [86] introduced an algorithm capable of highlighting features due to cocurvilinearity and proximity. The authors suggested a maximum likelihood directional vector field describing the contribution of a single unit-length edge element to its neighborhood in terms of length and direction. This field could reveal perceptual phenomena such as endpoint formations and straight lines. The output of the algorithm is a set of oriented features with an associated strength reflecting its saliency. In the assignment of probabilities to the field, the authors considered

two short edge segments, *perpendicular* to each other and apart. This scenario was a middle point between a choice of a connection by a sharp junction and a connection by a smooth curve. In addition, angles $\ll 90^\circ$ suggested a corner, while angles $\gg 90^\circ$ suggested a smooth connection.

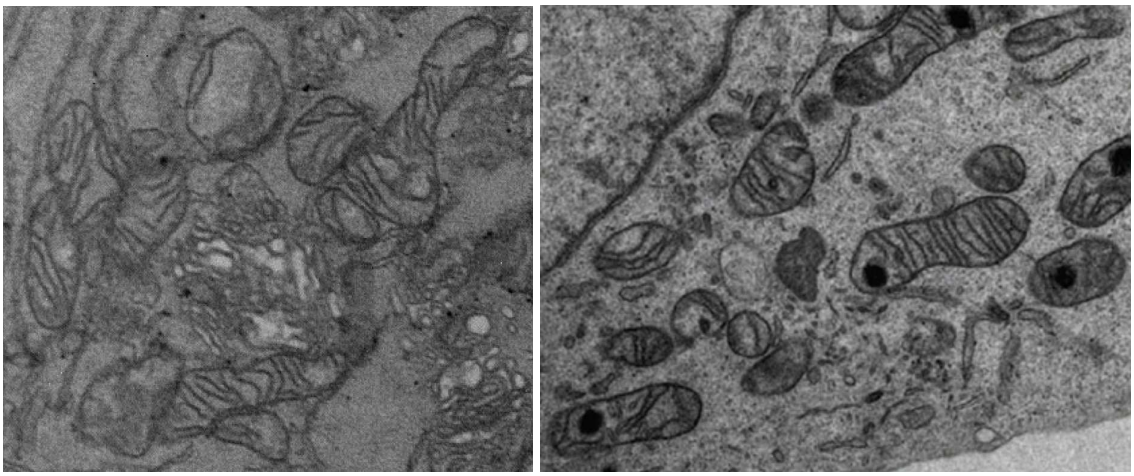
Heitger et al. [90] presented a computational model of a contour mechanism first identified by neurophysiological methods in monkey visual cortex. The scope was the definition of occluding contours in static monocular images. The authors considered the formation of occluding contours to be a statistical process. They assumed that foreground and background are statistically independent. The authors introduced a concept of *ortho* curvilinear grouping. Ortho grouping can be applied to terminations of the background, which tend to be orthogonal to the occluding contour. This grouping model also identifies the direction of figure and ground at such contours.

Williams and Jacobs [225] introduced an algorithm and the representation-level theory of illusory contour shape and salience. Their model was derived from an assumption that the prior probability distribution of boundary completion shape can be modeled by a random walk in a lattice the points of which are positions and orientations in the image plane. The authors characterized the distribution of completion shapes using the mathematical device of a particle undergoing a stochastic motion. Their random walk model embodied the Gestalt principles of proximity and good continuation, which originate in the statistics of the environment in which human visual systems evolved.

Grossberg and Mingolla [83] investigated the neural dynamics of form perception and boundary completion. In their paper, the authors described the use of a real-time visual processing theory to analyze real and illusory contour formation and contour and brightness interactions. Two parallel contour-sensitive processes interacted to generate brightness, color and form estimates. A boundary contour process was sensitive to orientation and amount of contrast, but not to the direction of contrast in scenic edges. It synthesized boundaries sensitive to the global configuration of scenic elements. A feature contour process was insensitive to orientation, but sensitive to both amount of contrast and to direction of contrast. The proposed model of illusory contour formation involved repeated convolution with a large-kernel filter. This kernel resembled the one of Williams and Jacobs [225], but did not represent the Green's function of a stochastic process. Grossberg and Mingolla's network was complex and its convergence properties were difficult to analyze. Part of this complexity stemmed from their desire to model, in a comprehensive way, the many different forms of stimulus which can elicit illusory contours. No other model, including that of Williams and Jacobs, attempted to be this comprehensive [225]. Although, contour integration and shape completion problem is not the main topic of this chapter, we discuss the applicability of our *Projectivity* measure to the grouping of co-linear and co-circular line segments in Section 4.9.

4.2 Feature Extraction Methods

Our motivation to explore the role of projections in line grouping originates in the field of nano-biophotonics and imaging of subcellular regions. Mitochondria form an important category of membrane enclosed, on average 200nm large *organelles* which reside inside every living cell. Mitochondrial morphology depends on the type of biological tissue and undergoes changes during induced or naturally occurring biochemical processes [199]. This fact accounts for the vast range



(a) (b)

Figure 4.2.1: Comparison of TEM images of clustered versus well-separated mitochondria. (a) A TEM image provided by our biomedical partners from RCSI containing clustered mitochondria in DU-145 human prostate cell. (b) An electron micrograph from [199] showing relatively well separated mitochondria in human HeLa cells.

of mitochondrial shapes and textures, and challenges a unified approach to localization and segmentation of these structures in electron microscopy images.

In many cases, transmission electron microscope images of mitochondria show characteristic quasi-linear structural elements (Fig. 4.1.1(c)-(d)) which can be converted to the proper line segments. Therefore a structural approach to the recognition of mitochondria exhibiting a predominantly linear or quasi-linear pattern may benefit from our *Projectivity* inference. However, we also make some assumptions when processing the subcellular microscopy images. We assume that the mitochondrion is always present, and that we know the number of mitochondria *a priori*. Because we are using the k-means clustering algorithm, this fact requires setting the value for the number of clusters k. We also assume that the inner folding patterns are consistent and always present. For abnormal cases of mitochondrial morphology at the later stages of apoptosis, the inner folding patterns change and our proposed method does not work.

Fig. 4.2.1 shows two different subcellular image settings. Fig. 4.2.1(a) shows a challenging feature extraction scenario, as in our case with clustered and deformed organelles. Well-separated mitochondria with distinctive inner membrane folding and shapes are shown in Fig. 4.2.1(b). In such a setting, a pattern recognition and segmentation approach could focus, for example, on the boundaries or the shape features of mitochondria.

Geometric descriptors such as, for example, Histograms of Oriented Gradients [43] have been proven to be highly efficient in the human detection task. The feature vector for the HOG implementation is obtained by dividing the image into small connected regions and for each region compiling a histogram of gradient directions for the pixels within the region. Haar-like features are based on the gray intensity and are insufficient to describe an object's texture. As noted by Smith [193], Haar features and HOG are inefficient at detecting highly deformable objects such as biological cells. Haar- and HOG-based weak learners are not particularly well-suited to this task

because they depend on reliable cues defined at precise locations, such as the nose appearing in the center of a face. To overcome these limitations, Smith et al. [193] developed a Ray feature detector for well-separated mitochondria with irregular shapes. Rays exploit the observation that while we cannot predict the precise locations of characteristic image features for highly deformable objects, we can predict their locations relative to one another, or relative to certain other locations. Although Ray descriptors were specifically designed to describe structures with strong edges, such as mitochondria, they achieve their best performance when used in combination with gray value histograms [132]. Similarly, Local Binary Patterns have been extensively used for describing textures, one of their features being their robustness to illumination changes. However, as shown by Cetina et al. [30], they do not perform well in electron microscopy images possibly because the latter have a very strong background noise. An object's texture information can be also obtained through the statistics of Gray Level Co-occurance Matrix. Combining GLCM and Haar features can improve the descriptive capability.

It has been shown empirically that trying to segment subcellular structures in neural tissue images, for example, using only geometric or textural features, is not very effective [30]. The Radon descriptors, which also involve computation of line segments, were proposed as a solution to this problem. Such feature descriptors are designed to leverage both the texture and the geometric information present in such images to segment mitochondria. However, in the case of both deformed and clustered mitochondria, as shown in Fig. 4.2.1(a), these approaches may not work. The task of finding mitochondria here is made even more difficult by the presence of many other irregular objects. It can be seen from Fig. 4.2.1(a) that at least some of mitochondrial inner membrane folding can be well-approximated by line segments. In this case, the Haar [103] or Radon [113] features can be used to enhance the membranes. Thus, the guiding idea of our approach is to extract and to group the line segments, and thus to obtain a localization marker which can be used in the subsequent outer mitochondrial contour extraction. Also, extracting and connecting line segments is a less difficult and less computationally expensive task than detecting the mitochondrion regions using, for example, the texture and boundary feature detectors discussed.

4.3 Grouping Objective

We outline the basis for the grouping methodology presented in this work by providing an example. Fig. 4.3.1(a) shows two synthetic objects C and D with intrinsic linear features and dynamic shape. All line segments belonging to object C feature different lengths, orientations and density. One possible way to recover the original shape of either object is to link closest line segments by their spanned orthogonal projections. In Fig. 4.3.1(b), we proceed from the "head" line segment to the "tail" line segment of object C, while iteratively constructing one-sided orthogonal projections of one line segment onto its next neighbor. In Fig. 4.3.1(c) we construct double-sided or pairwise orthogonal projections and observe that the total connected area yields a more accurate estimate of the original shape of both objects. Therefore, for the aim of grouping we do not pose any constraints on the length, orientation or density of line segments, but rather consider the combination of the following grouping cues:

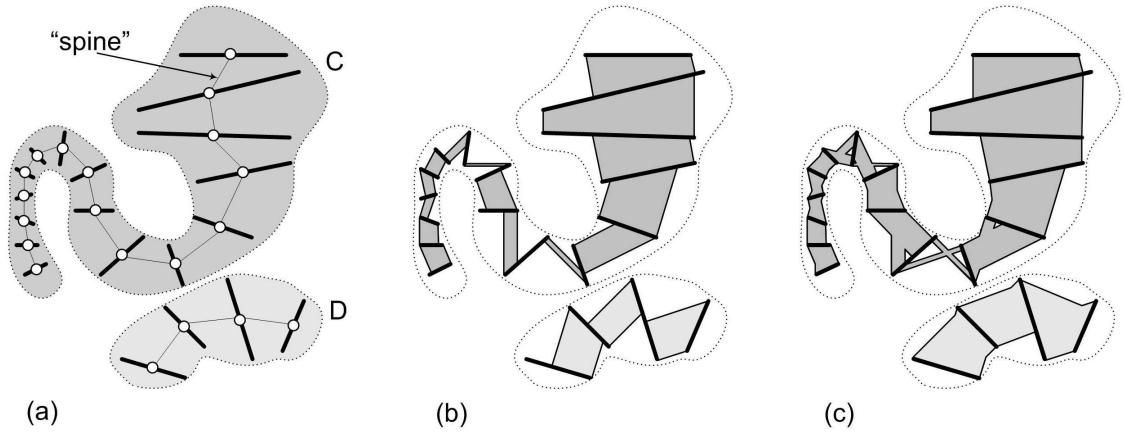


Figure 4.3.1: **(a)** Two synthetic objects C and D, composed of groups of line segments. **(b)** Case of one-sided orthogonal projections between two closest line segments. **(c)** Case of double-sided orthogonal projections. Objects C and D aim to model mitochondrial morphology.

(i) ***Projectivity***: Line segments form a cluster with pairwise orthogonally projected entities.

(ii) ***Proximity***: Line segments in a cluster follow some proximity principle in Euclidean space.

4.4 Similarity Measures

A line segment X can be described by a vector notation $\mathbf{x}$ and can be sufficiently parameterized by its two end points s_x and e_x: $\mathbf{x} = [s_x, e_x]$. From this representation, the secondary parameters such as center c_x, length l_x and direction α_x of a line segment, and the angle between two line segments $\Delta_\alpha : \angle\mathbf{x}, \mathbf{y}$, can be directly inferred by means of elementary trigonometry and algebra. The basic relationships between two line segments are shown in Fig. 4.4.1(a). We define the distance D between two line segments, X and Y, as the Euclidean distance between the centers c_x and c_y:

$$D(X,Y) : D(c_x, c_y) = |c_x - c_y| \cdot |c_x - c_y|^\top = \|\mathbf{c}_{xy}\| \tag{4.1}$$

The "spine" Π of the detected cluster can be represented by a polygon with vertices $c_m, c_{m+1}, ..., c_M$ given by the centers of elementary line segments. The polygonal length between the start c_1 and the end c_M of a polygon is given as:

$$\Pi(c_1, c_M) = \sum_{m=1}^{M-1} D(c_m, c_{m+1}) \tag{4.2}$$

The vector $\mathbf{c}_{xy}$ describes the polygon edge (c_x, c_y) and determines the angle $\phi_{xy} : \angle\mathbf{x}, \mathbf{c}_{xy}$. The complete set of parameters used in all of the follow calculations comprises: $\{s, e, c, \alpha, \Delta_\alpha, \phi\}$.

4.4.1 Orthogonal Projections

Let $P_X Y$ be named the *left-projection* and represent the orthogonal projection of line segment Y onto line segment X. The *right-projection* is the orthogonal projection of line segment X onto line

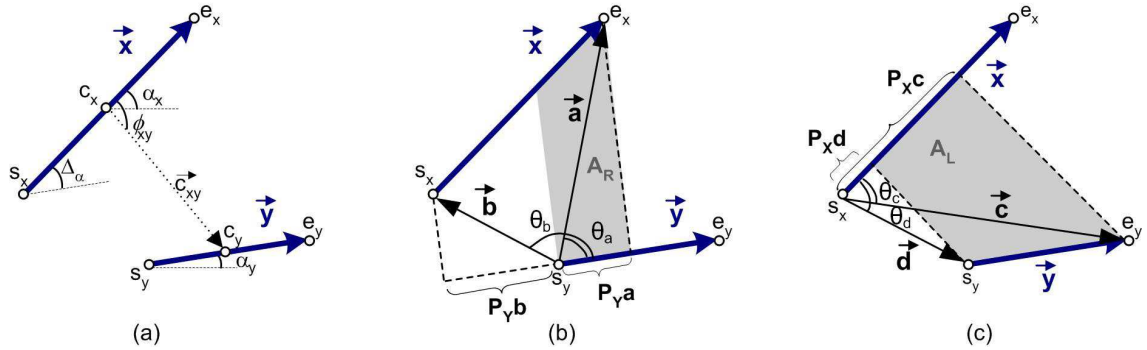


Figure 4.4.1: Variables involved in the computation of a pairwise similarity measure for grouping two line segments. (a) Notations. (b) Right-orthogonal projection. (c) Left-orthogonal projection.

segment Y and is denoted by P_YX. In order to obtain P_YX in closed form, we define two supporting vectors $\mathbf{a}$ and $\mathbf{b}$ (see Fig. 4.4.1(b)) as follows: $\mathbf{a} = [s_y, e_x]$ and $\mathbf{b} = [s_y, s_x]$. The angles θ_a and θ_b are directly related to the dot product between the corresponding vectors: $\cos\theta_a = (\mathbf{a} \cdot \mathbf{y}^T)/(\|\mathbf{a}\| \cdot \|\mathbf{y}\|)$ and $\cos\theta_b = \mathbf{b} \cdot \mathbf{y}^T/(\|\mathbf{b}\| \cdot \|\mathbf{y}\|)$. The orthogonal projections of vectors $\mathbf{a}$ and $\mathbf{b}$ onto line segment Y therefore are $P_Ya = \|\mathbf{a}\| \cdot \cos\theta_a$ and $P_Yb = \|\mathbf{b}\| \cdot \cos\theta_b$. As an analogy, in order to obtain the P_XY projection, we define two additional support vectors $\mathbf{c}$ and $\mathbf{d}$ as: $\mathbf{c} = [s_x, e_y]$ and $\mathbf{d} = [s_x, s_y]$. Let us introduce a binary cost function Ω:

$$\Omega = \begin{cases} 1, & \text{if } \cos\theta > 0; \\ 0, & \text{otherwise.} \end{cases} \tag{4.3}$$

Then the orthogonal projections P_YX and P_XY are:

$$\begin{aligned} P_YX &= \mid P_Ya \cdot \Omega_a - P_Yb \cdot \Omega_b \mid, \\ P_XY &= \mid P_Xc \cdot \Omega_c - P_Xd \cdot \Omega_d \mid . \end{aligned} \tag{4.4}$$

The projected areas A_L and A_R, spanned by the left- and the right- projections can be obtained as:

$$\begin{aligned} A_R &= \|\mathbf{a}\| \cdot \sin\theta_a \cdot P_YX - 1/2 \cdot (P_YX)^2 \cdot \tan\Delta_\alpha \\ A_L &= \|\mathbf{c}\| \cdot \sin\theta_c \cdot P_XY - 1/2 \cdot (P_XY)^2 \cdot \tan\Delta_\alpha \end{aligned} \tag{4.5}$$

The proposed Projectivity grouping cue is defined by the orthogonal projections in (4.4) and is further factored into the final similarity model presented next.

4.4.2 Projection-to-Distance Ratio

We are interested in expressing a similarity between two line segments X and Y as a combination of their pairwise orthogonal projections P as well as their center-to-center distance D. Initially, we reserve higher similarity scores for higher values of P. In order to factor D into the joint similarity expression, we consider now the situation where we are given a set of three parallel and equal length line segments X, Y and Z with equal orthogonal projections $P_XY = P_YZ = P_XZ$ such that $D(X, Y) < D(X, Z)$. Here, we would like to assign a higher similarity score to the closer-lying pair of line segments (X, Y) and a lower score to the more remote pair (X, Z). This relational tendency of maximizing the outcome with the increase of P and the simultaneous reduction of D can generally

be represented by a variety of models. In this work, we refrain from using a linear $P - D$ model in order to avoid negatively valued similarities. Instead, we apply a non-linear inverse model in the form of a *Projection-to-Distance* ratio which we denote as the R similarity ratio. With the notations for the left- and right-orthogonal projections defined above, the combined similarity measures are defined as:

$$R(X,Y) = \frac{P_X Y}{D(X,Y)}, \qquad R(Y,X) = \frac{P_Y X}{D(X,Y)} \tag{4.6}$$

Generally, for a similarity measure R to be considered a *metric* on the finite measurement space E it has to satisfy the following four conditions for every $X, Y, Z \in E$ [9, 152]:

- $R(X,Y) = \infty$ if $X = Y$, (Identity Axiom)
- $R(X,Y) \geq 0$, (non-negative and real valued)
- $R(X,Y) = R(Y,X)$, (Symmetry Axiom)
- $R(X,Y) \leq R(X,Z) + R(Y,Z)$ (Triangular Inequality)

We examine the R ratio for compliance with the above conditions. The identity axiom is the declaration of self-similarity, which implies that the distance measure $D(X,X)$ is zero, but the joint similarity measure $R(X,X)$ is an infinitely large number. It can be seen from Figure 4.4.1, that if $X = Y$, then vectors $\mathbf{a} = X$, $\mathbf{b} = 0$, $\mathbf{c} = Y$, $\mathbf{d} = 0$ and $\cos\theta_{(a,c)} = 1$. Furthermore, as $P(X,X) = \|X\|$ and $P(Y,Y) = \|Y\|$, we arrive (supported by continuity arguments) at the self-similarity expression:

$$R(X,X) = \lim_{D\to 0} \frac{P(X,X)}{D(X,X)} = \lim_{D\to 0} \frac{\|X\|}{D} = \infty \tag{4.7}$$

We therefore infer that the identity property is satisfied because $R(X,Y) = \infty$ for the case of identical line segments X and Y.

The second axiom holds on the basis of the definitions of orthogonal projections in (4.4) which are defined as non-negative and real valued numbers, $R(X,Y) \geq 0$.

By providing a counter example in Fig. 4.4.2 and Table 4.4.1, we demonstrate that the symmetry condition is not satisfied as the left projection $P_X Y$ does not equal the right projection $P_Y X$. This further translates to $R(X,Y) \neq R(Y,X)$. The weak Triangular Inequality does not hold for all $X, Y, Z \in E$ as in the example of $R(X,Y) \nleq R(X,Z) + R(Y,Z)$ (Fig. 4.4.2 and Table 4.4.1).

Table 4.4.1: Supplementary Data for the Set of Line Segments in Fig. 4.4.2

	(X,Y)	(Y,X)	(X,Z)	(Z,X)	(Y,Z)	(Z,Y)
P	4.8	5.75	3.55	3.6	0	0.3
D	5.2	5.2	6.6	6.6	6.32	6.32
R	0.92	1.1	0.54	0.55	0	0.05

To conclude, two out of four necessary conditions are not fulfilled and thus the R-ratio is not metric. Because the R-ratio is not metric, we conclude that it is also a non-Euclidean similarity. Caution should be exercised in the reverse case: if a similarity measure has been proven to be metric, then it still remains to prove that it is Euclidean [64].

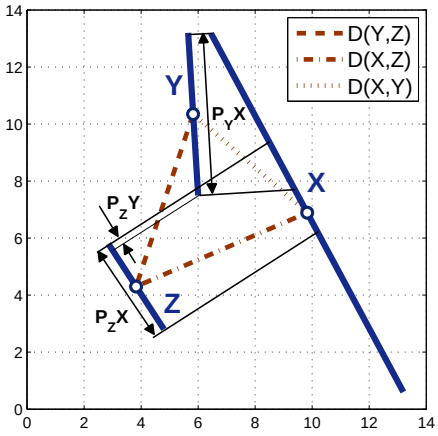


Figure 4.4.2: An example of a set of line segments {X,Y,Z} that meet neither the symmetry nor the triangle inequality metric requirements for the *R* similarity measure.

Table 4.4.2: Notations used throughout this chapter.

$\mathbf{P}$	asymmetric matrix containing pairwise orthogonal projections
$\mathbf{D}$	symmetric matrix containing pairwise Euclidean distances
$\mathbf{R}$	asymmetric similarity matrix
$\mathbf{R}_{sym}$	symmetric part of $\mathbf{R}$
$\mathbf{R}_{skew}$	skew-symmetric part of $\mathbf{R}$
$\mathbf{G}$	Gram matrix used in the kernel-PCA framework
$\mathbf{A}$	dissimilarity matrix with elements a_{ij}
$\mathbf{K}$	matrix containing squared dissimilarities a_{ij}^2
N	number of line segments
$\mathbf{p}_N$	N-vector of ones
$\mathbf{I}$	$N \times N$ identity matrix
$\mathbf{Q}$	projection matrix on the orthogonal complement of $\mathbf{p}_N$
$\mathbf{S}^c$	centralized pseudo-covariance matrix
$\mathbf{U}$	matrix containing eigenvectors as columns
$\mathbf{\Lambda}$	matrix containing eigenvalues on the main diagonal
λ_{min}	minimal eigenvalue of $\mathbf{\Lambda}$
$\tilde{\mathbf{S}}^c$	embedded centralized pseudo-covariance matrix
$\mathbf{X}$	kernel-PCA based feature vectors
L	number of dimensions retained after the dimensionality reduction step
k	number of k-means clusters
Nn	number of noise-related line segments
Ns	number of signal-related line segments
ϵ	deviation from Euclideaness

4.4.3 Similarity Matrix and its Properties

For the grouping of N line segments, we construct a square $N \times N$ similarity matrix $\mathbf{R}$ which captures the pairwise relationships between all N line segments. The corollary of the violated

symmetry axiom is the asymmetric structure of the similarity matrix $\mathbf{R}$ with the following entries for row i and column j indices:

$$\mathbf{R}(i,j) = \begin{cases} R(X,Y), & \text{if } i < j, \text{ left-projections;} \\ R(Y,X), & \text{if } i > j, \text{ right-projections;} \\ \infty, & \text{if } i = j, \text{ diagonal.} \end{cases} \tag{4.8}$$

As can be seen from (4.8), the upper triangular part of matrix $\mathbf{R}$ intrinsically contains left-projections $P_X Y$ while the lower triangular part contains right-projections $P_Y X$ as defined in (4.4). The matrix in (4.9) is an example which demonstrates the inherent asymmetry property and depicts the relationships between four line segments v_1, v_2, v_3 and v_4.

$$\mathbf{R} = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} \infty & 0.9073 & 0.5867 & 0 \\ 0.5145 & \infty & 1.4202 & 0 \\ 0.6199 & 1.3868 & \infty & 0.0163 \\ 0 & 0 & 0 & \infty \end{pmatrix} \end{array} \tag{4.9}$$

In addition to the ∞-valued diagonal, the presence of zero-valued or "missing" elements is also peculiar. A zero entry in matrix $\mathbf{R}$, such as $\mathbf{R}(v_4, v_3) = 0$ indicates that the orthogonal projection of line segment v_3 onto line segment v_4 simply does not exist. This feature may lead to a high percentage of zero-valued entries, such as 85% in the example shown in Fig. 4.4.3(b). This plot also demonstrates that the non-zero entries of $\mathbf{R}$ are randomly distributed. We highlight that this apparent randomness is induced through the random permutation of spatial locations of line segments before the construction of $\mathbf{R}$. In order for the similarity matrix $\mathbf{R}$ to acquire a *block-diagonal* structure, the spatial coordinates of line segments have to be sorted either in ascending or descending order. Our decision to refrain from this practice has been influenced by Novembre and Stephens's work [159]. The authors point out that such an ordered spatial sampling introduces mathematical artefacts in the exploratory data analysis and may impair the interpretation of results. However, in applications where the emphasis is put on the computational benefits of the block-diagonal structure [166], the uniform sampling scheme is clearly an option.

Because of its asymmetry, $\mathbf{R}$ may have complex eigenvalues and eigenvectors and cannot be generally considered being a PSD matrix. This condition is necessary for a *loss-free* Euclidean embedding of a similarity measure [28, 64] and for the feasibility of factorization-based grouping approaches [104, 166]. The asymmetric and non-Euclidean properties of $\mathbf{R}$ also question the applicability of centroid-based grouping methods such as k-means and Fuzzy C-Means [64]. The latter assume that a similarity matrix contains data vectors rather than asymmetric relations. As a consequence, the computed center of gravity of a centroid may not have a valid geometric interpretation in non-Euclidean space. Thus, a symmetrization step is necessary when pairwise similarity relations are inherently asymmetric as in our projection-based similarity model. The resulting similarity matrix $\mathbf{R}$ is then inherently asymmetric, and there are several possibilities to transform $\mathbf{R}$ into a symmetric matrix. According, for example, to Constantine and Gower [38], an asymmetric

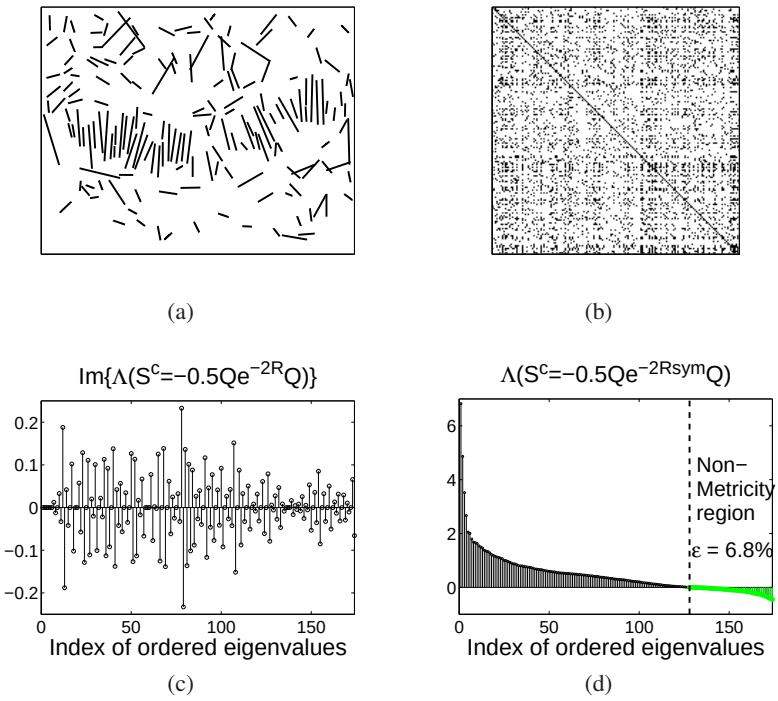


Figure 4.4.3: Spectral properties of matrix **R**. **(a)** Set of N=174 line segments extracted from the image in Fig. 1.1(c). **(b)** Similarity matrix **R** containing 85% zero-valued entries which are shown in white. **(c)** Imaginary part of the complex eigenspectrum of the **G**, constructed with the asymmetric **R**. **(d)** Plot of eigenvalues of **G**, constructed with the symmetrized matrix $\mathbf{R}_{sym}$. Here, the deviation from Euclideaness is coded by negative eigenvalues and amounts to $\epsilon = 6.84\%$.

matrix **R** can be decomposed into a symmetric $\mathbf{R}_{sym}$ and a skew-symmetric $\mathbf{R}_{skew}$ as follows:

$$\begin{aligned} &\mathbf{R} = \mathbf{R}_{sym} + \mathbf{R}_{skew}, \\ &\mathbf{R}_{sym}(i,j) = \mathbf{R}_{sym}(j,i) = \tfrac{1}{2}(\mathbf{R}(i,j) + \mathbf{R}(j,i)), \\ &\mathbf{R}_{skew}(i,j) = \tfrac{1}{2}(\mathbf{R}(i,j) - \mathbf{R}(j,i)) \end{aligned} \tag{4.10}$$

In this formulation, every element of $\mathbf{R}_{sym}$ equals the arithmetic mean between r_{ij} and r_{ji}. The skew symmetric part $\mathbf{R}_{skew}$ is frequently disregarded in the subsequent calculations, but it is worth noting that this practice results in a loss of information related to correct cluster assignments. The symmetrizing transformation in (4.10) as stated by Roth et al. [176], is a necessary condition for loss-free Euclidean embedding by means of the Constant Shift Embedding procedure. Let us denote a dissimilarity (distance) matrix by **A**. We can then write the combined dissimilarity mapping and symmetrization as $\mathbf{R} \rightarrow \mathbf{R}_{sym} \rightarrow \mathbf{A}$.

Alternatively, it is also possible to perform the dissimilarity mapping first and to symmetrize **A** afterwards. Let us denote the symmetric part of the dissimilarity matrix by $\mathbf{A}_{sym}$. Then the latter is obtained through the transformation $\mathbf{R} \rightarrow \mathbf{A} \rightarrow \mathbf{A}_{sym}$. However, by taking the averaging symmetrizing transformation in (4.10) as given, the order of operations $\mathbf{R} \rightarrow \mathbf{R}_{sym} \rightarrow \mathbf{A}$ and $\mathbf{R} \rightarrow \mathbf{A} \rightarrow \mathbf{A}_{sym}$ is not commutative in general, except for linear dissimilarity mapping.

To exclude the possibility that the non-linearity of the inverse exponential operation on a non-metric **R** translates into a metric **A**, we revisit the example provided in Fig. 4.4.2. We define two different dissimilarity transformations A_1 and A_2, and the A_{2sym} which is the symmetric part of A_2.

The results from Table 4.4.3 show that neither transformation leads to a valid Triangular Inequality, such that $A_1(Y,Z) < A_1(X,Y) + A_1(X,Z)$ or $A_{2sym}(Y,Z) < A_{2sym}(X,Y) + A_{2sym}(X,Z)$.

Table 4.4.3: Dissimilarity Mapping for the Set of Line Segments in Fig. 4.4.2

	(X,Y)	(Y,X)	(X,Z)	(Z,X)	(Y,Z)	(Z,Y)
P	4.8	5.75	3.55	3.6	0	0.3
D	5.2	5.2	6.6	6.6	6.32	6.32
R	0.92	1.1	0.54	0.55	0	0.05
R_{sym}	1.01		0.545		0.025	
$A_1 = e^{-R_{sym}}$	0.3642		0.5798		0.9753	
$A_2 = e^{-R}$	0.4	0.33	0.583	0.577	1	0.95
A_{2sym}	0.3657		0.5798		0.9756	

Since there are no other restrictions, we first decompose $\mathbf{R}$ as $\mathbf{R} = \mathbf{R}_{sym} + \mathbf{R}_{skew}$ and then apply the inverse exponential mapping on the symmetric $\mathbf{R}_{sym}$ to obtain a valid matrix $\mathbf{A}$.

We denote the asymmetric matrix containing pairwise orthogonal projections as $\mathbf{P}$, and the matrix containing pairwise Euclidean distances between the centres of line segments as $\mathbf{D}$. The asymmetric matrix $\mathbf{R}$ is constructed using element by element division $\mathbf{P}/\mathbf{D}$. Using the defined notations, the processing pipeline for the dissimilarity mapping can be written as:

$$\{\mathbf{P},\mathbf{D}\} \rightarrow \mathbf{R} = \frac{\mathbf{P}}{\mathbf{D}} \rightarrow \mathbf{R}_{sym} \rightarrow \mathbf{A} = e^{-\mathbf{R}sym} \tag{4.11}$$

Due to the symmetry of $\mathbf{D}$, the symmetrization of $\mathbf{R}$ according to (4.10) is also equivalent to the approach of symmetrizing $\mathbf{P}$ at first and constructing $\mathbf{R}$ afterwards. The skew-symmetric matrix $\mathbf{R}_{skew}$ contributes to the existence of a complex eigenspectrum of the pseudo-covariance[2] matrix $\mathbf{G}$ as can be seen in Fig. 4.4.3(c). After symmetrization, the deviation from Euclideaness is generally coded in the negative part of the eigenspectrum [117, 164]. The symmetrized matrix $\mathbf{R}_{sym}$ yields real, but not necessarily positive eigenvalues as can be seen in Fig. 4.4.3(d). These observations fit well with previous studies which also emphasize the fact that data coded in the complex part which describes the departure from symmetry, and in the negative part of the eigenspectrum, can in principle be informative. Therefore any reduction of the eigenspectrum results in loss of information [117, 164, 176]. We examine the resulting $\mathbf{R}_{sym}$ and $\mathbf{A}$ for the example given in (4.9):

$$\mathbf{R}_{sym} = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} \infty & 0.7109 & 0.6033 & \mathbf{0} \\ & \infty & 1.4035 & \mathbf{0} \\ & & \infty & 0.0081 \\ & & & \infty \end{pmatrix} \end{array} \tag{4.12}$$

$$\mathbf{A} = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} \mathbf{0} & 0.4912 & 0.5470 & \mathbf{1} \\ & \mathbf{0} & 0.2457 & \mathbf{1} \\ & & \mathbf{0} & 0.9919 \\ & & & \mathbf{0} \end{pmatrix} \end{array} \tag{4.13}$$

[2]The notation $\mathbf{G}$ is used in the kernel-PCA framework which is described in Section 2.2.6.4

This turns out to be a case where Triangular Inequality holds for every triplet in **A**. Therefore this example seems to result in a *metric* dissimilarity matrix **A**. There is, however, a chance that the effect of decimal rounding accounts for the latter observation. To investigate this assumption further, we analyze the eigenspectrum of the corresponding centralized Gram matrix **G** [176]. The spectral decomposition of **G** returns a set of four eigenvalues $\lambda = [6.9695, 1.7128, 0.2795, -2.15e-012]$. Even though it is very small, the negative-valued entry $\lambda_{min} = -2.15e-012$ indicates that **G** is not positive semidefinite and therefore is not derived from squared Euclidean distances.

The occurrence of the negative spectra of **G** can be traced back to the fundamentally non-metric behavior of orthogonal projections P, and should not be only viewed as a result of a competition between two features (in our case P and D) in a ratio model [9, 118]. Based on the data in Table 4.4.3 and with the symmetrization step given by (4.10), one can easily see that $P_{sym}(X,Y) \nleq P_{sym}(X,Z) + P_{sym}(Z,Y)$. The violated Triangular Inequality verifies the non-metricity of P_{sym}, which further propagates into the ratio model.

4.5 Grouping Strategy

We have already highlighted the reasons for choosing the unsupervised machine learning tool of spectral clustering in this thesis. In particular, we chose the embedded kernel-PCA [176], and the integration of perceptual grouping constraints into the similarity model as the key elements of the approach to the localization and segmentation of mitochondria in microscopic images. When applied, this strategy results in a reconstructed low-dimensional feature space where the classification follows and where the inherent and meaningful structures in the input data are represented by visible clusters of feature points. The next step is then to partition the feature space into the appropriate clusters.

A key consideration for the selection of *any* grouping method is how it treats asymmetric non-Euclidean similarity data, arbitrary cluster shapes and possibly varying cluster densities. For instance, Single Linkage Hierarchical Clustering which is a nearest neighbor grouping method, operates on a symmetric, but not necessarily a metric dissimilarity matrix, and may result in chained and branched [9] clusters. A number of partitioning methods deal separately either with non-Euclidean data or with non-convex cluster shapes. The CURE (Clustering Using REpresentatives) [84] algorithm is capable of detecting arbitrarily shaped and sized clusters, but assumes a Euclidean metric. In contrast, the NERF (Non Euclidean Relational Fuzzy) C-Means algorithm by Hathaway and Bezdec [89] introduces the *β-spread* transformation of symmetric non-Euclidean PSD similarity matrices, but as the "-means" term suggests, yields spherical or ellipsoidal clusters.

In 2007, the Affinity Propagation (AP) algorithm was introduced by Frey and Dueck in *Science* magazine [70]. It is a machine-learning clustering approach which can directly be applied on asymmetric, non-metric and sparse matrices. Despite this unique generality, the original form of AP assumes more or less convex-shaped clusters. Several improvements have been proposed in this regard, for instance the soft-constrained AP (SCAP) algorithm in [123], which was successfully tested on interlocking "U" shaped clusters. Yet, none of the mentioned algorithms has a nearly as well-defined and acknowledged framework of noise analysis and reduction as that of eigenvectors-based clustering methods [176].

Table 4.5.1 compares two selected grouping strategies I and II. Both accept a similarity matrix $\mathbf{R}$ as a single precursor and further symmetrize and transform it into the distance-related *dissimilarity* matrix $\mathbf{A}$. While single linkage hierarchical clustering (I) works directly on the distance matrix $\mathbf{A}$, spectral (II) methods combined with k-means clustering, frequently seek vectorial representation of $\mathbf{A}$ and embed the non-metric data into Euclidean space prior to the eigendecomposition [164, 175]. In this regard, the flow of events encapsulates three major steps: formation of a *pseudo*-Covariance matrix $\mathbf{S}^c$, Euclidean embedding of $\mathbf{S}^c$ and then PCA on the embedded $\tilde{\mathbf{S}}^c$.

Table 4.5.1: Comparison of two selected grouping strategies

	Transformation step	I	II
1	Similarity Matrix	$\mathbf{R}$	$\mathbf{R}$
2	Symmetrizing Transformation	$\mathbf{R}_{sym}$	$\mathbf{R}_{sym}$
3	Dissimilarity Mapping	$\mathbf{A}$	$\mathbf{A}$
4	Kernel Matrix		$\mathbf{K}$
5	*pseudo*-Covariance Matrix		$\mathbf{S}^c$
6	Euclidean Embedding		$\tilde{\mathbf{S}}^c$
7	kernel-PCA		$\tilde{\mathbf{S}}^c = \mathbf{U}\Lambda\mathbf{U}^T$
	$\Rightarrow$ vectorial representation		$\mathbf{X} = \mathbf{U}\sqrt{\Lambda}$
8	non-Nearest Neighbor Grouping (k-means)		✓
9	Nearest Neighbor Grouping (Single Linkage Hierarchical)	✓	

As a preliminary to understanding the grouping strategy II outlined in Table 4.5.1, we introduce a centralized dot product (Gram) matrix $\mathbf{S}^c$ such as:

$$\mathbf{S}^c = -\tfrac{1}{2}\mathbf{QKQ}, \mathbf{Q} = \mathbf{I}_N - \tfrac{1}{N}\mathbf{p}_N\mathbf{p}_N^T \tag{4.14}$$

where $\mathbf{K} : [a_{ij}^2]$ is a matrix of squared dissimilarities, $\mathbf{I}_N$ is a $N \times N$ identity matrix and $\mathbf{p}_N = (1, 1, ..., 1)^T$ is an N-vector of ones. Therefore, $\mathbf{Q}$ is the projection matrix on the orthogonal complement of $\mathbf{p}_N$. For the case of squared Euclidean dissimilarities, matrix $\mathbf{K}$ is a Gram matrix and matrix $\mathbf{S}^c$ is also a PSD covariance matrix – a sufficient indication of the Euclideaness [176] of the data. The selected dissimilarity mapping in (4.11) has the advantage of emphasizing small-scale dissimilarities a_{ij} by suppressing the effect of large distances and thus potentially the noise. The resulting dissimilarity values are distributed in the range $[0, 1]$. The fact of using a Gaussian kernel provides us with the advantage of performing clustering in non-linear space where the linear boundary between two partitions can be better found while applying the linear k-means algorithm.

To quantify the deviation from Euclideaness, we adopt a measure ε suggested by Podani and Miklos in [167] as the ratio between the sums of positive λ_p and negative λ_n eigenvalues $\times$ 100%:

$$\epsilon = \frac{\sum \lambda_p}{|\sum \lambda_n|} \cdot 100\% \tag{4.15}$$

In order to enable a loss-free embedding of dissimilarity data into a vector space, the spectral embedding approaches seek to alter $\Lambda(\mathbf{S}^c)$ by analyzing its departure from Euclideaness, ϵ. Relatively small values of ϵ are frequently considered as being an indicator for non-informative content

[164]. Therefore, approaches which remove noise-related dimensions represented by ϵ also reason that this procedure efficiently resolves the problem of noise reduction.

4.6 Metric Embedding and kernel-PCA

The negative part of the eigenspectrum, in some cases, can represent significant relations in the input data and thus its removal can result in the loss of information [164]. The disadvantage can be avoided by transposing $\Lambda(\mathbf{S}^c)$ into the positive range. This is a guiding idea for approaches such as Non-metric Multidimensional Scaling as well as for CSE (Constant Shift Embedding), introduced by Roth et al. [176, 175]. In particular, in the CSE framework the complete eigenspectrum is raised by the smallest negative eigenvalue λ_{min}. According to Roth et al. [176, 175], the following Lemma states that for any indefinite, real and symmetric matrix $\mathbf{S}^c$, a positive semidefinite matrix $\tilde{\mathbf{S}}^c$ can be derived by subtracting the smallest eigenvalue from its diagonal elements [175].

Theorem 1 *The kernel matrix $\boldsymbol{K}$ contains squared Euclidean distances, i.e., $\boldsymbol{K} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2$, if and only if $\boldsymbol{S}^c$ is positive semidefinite.*

Lemma 1 *Let $\tilde{\boldsymbol{S}}^c = \boldsymbol{S}^c - \lambda_{min}(\boldsymbol{S}^c)\boldsymbol{I}_N$, where $\lambda_{min}(\cdot)$ is the minimal eigenvalue of its argument. Then $\tilde{\boldsymbol{S}}^c$ is positive semidefinite.*

Furthermore, the reconstructed *Euclidean* vectorial data $\mathbf{X}$ is obtained through spectral decomposition of the embedded $\tilde{\mathbf{S}}^c = \mathbf{U}\Lambda\mathbf{U}^T$ and the subsequent re-scaling of eigenvectors such as $\mathbf{X} = \mathbf{U}\sqrt{\Lambda}$. This framework forms the basis for the kernel-PCA approach with a *quadratic* kernel matrix $\mathbf{K}$. The rows of $\mathbf{X}$ correspond to the multidimensional vectors $\mathbf{x}_i, i = 1, ..., N$. The dimensionality reduction step is carried out by taking the first L columns of $\mathbf{X}_L = \mathbf{U}_L\sqrt{\Lambda_L}$. The complete embedding and reconstruction path is given as:

$$\mathbf{A} \rightarrow \mathbf{K} : \{a_{ij}^2\} \rightarrow \mathbf{S}^c = -\frac{1}{2}\mathbf{QKQ} \rightarrow \mathbf{S}^c - \lambda_{min}(\mathbf{S}^c)\mathbf{I}_N \rightarrow \tilde{\mathbf{S}}^c = \mathbf{U}\Lambda\mathbf{U}^T \rightarrow \mathbf{X}_L = \mathbf{U}_L\sqrt{\Lambda_L} \quad (4.16)$$

The Gram matrix $\mathbf{S^c}$ essentially represents the dot product in the feature space. $\mathbf{S^c}$ plays the role of a pseudo-covariance matrix and is the starting point for both linear (PCA) and non-linear (kernel) PCA-based dimensionality reduction algorithms. The major mutual effect of CSE and kernel-PCA is that this approach, if used in conjunction with a shift-invariant clustering method such as k-means, results in the complete preservation[3] of the group structure in the embedding space [176].

4.7 Experiments on Synthetic Data Sets

In this section, we describe a set of synthetic experiments with the purpose of exploring the constellation of the reconstructed data in the kernel-PCA feature space, and of studying the effects of additive noise. The first experiment in Fig. 4.7.1 shows two synthetic objects comprised of line segments with different lengths and orientation, without structural noise, and clustered with different R models. The objective of this experiment is to show clustering results using different kernels.

[3] For non-metric data and without the embedding with CSE, the k-means algorithm will not return the correct clusters.

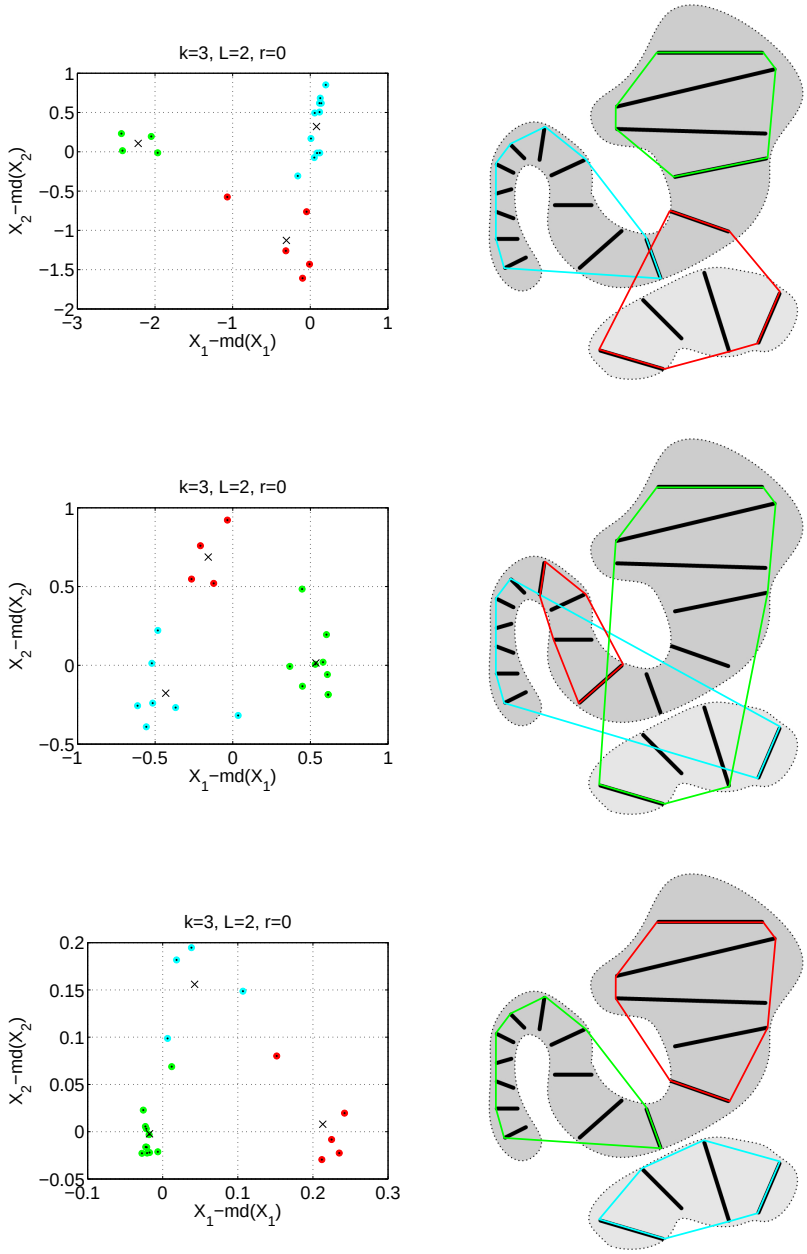


Figure 4.7.1: Different kernels and clustering results. Two objects of interest (from Fig. 4.3.1) consist of projected line segments without structural noise. Row1: $\mathbf{R} = \mathbf{P}/\mathbf{D}; \mathbf{A} = max(\mathbf{R}) - \mathbf{R}$; Row2: $\mathbf{R} = \mathbf{P}; \mathbf{A} = exp(-\mathbf{R})$; Row3: $\mathbf{R} = (1+\bar{\mathbf{P}})^2/(8.5+\bar{\mathbf{D}})$, where $\bar{\mathbf{P}} = \mathbf{P}/max(\mathbf{P}), \bar{\mathbf{D}} = \mathbf{D}/max(\mathbf{D})$. Here the parameters in the R-model are manually tuned such that k-means produces correct cluster assignments. Diagrams are best viewed in color.

The second synthetic experiment relies on the immersion of different signal structures into the stationary low-variance structural noise. Conversely, the structural noise in the last synthetic experiment is set to vary, while the signal structures remain unchanged.

Generation of noise-related line segments involves first, the generation of uniformly distributed center points, and second, the generation of end points drawn from the normal distribution. We relate the signal-to-noise ratio to the similarity matrices $\mathbf{R}$ of signal and noise structures and specifi-

cally to the average number of all non-zero (NZ) entries of corresponding matrices.

$$SNR = 10log_{10}\frac{\mu(\mathbf{R}_{signal,NZ})}{\mu(\mathbf{R}_{noise,NZ})}\,. \tag{4.17}$$

The results obtained from the second synthetic experiment can be observed in Fig. 4.7.2 and Fig. 4.7.3. The first row shows the four original synthetic structures with the number of signal-related line segments Ns, and the number of noise-related line segments Nn. The second row shows hierarchical cluster trees constructed using the dissimilarity matrix $\mathbf{A}$. One advantage of SLHC is that preselection of the final cluster number is not required, allowing a domain expert to analyze the resulting cluster hierarchy. However, the optimal number of clusters in SLHC is sensitive to the selection of the dendrogram cut-off value. The discussion on this topic as well as results of using SLHC on mitochondria images are presented in Section 4.7.2.

The spectral decomposition of the embedded $\tilde{\mathbf{S}}^c$ yields two interesting observations. Firstly, we observe a low variability of the $\tilde{\mathbf{S}}^c$ eigenspectrum (Fig. 4.7.2, Fig. 4.7.3 Row3), where the leading eigenvectors account only for a small cumulative percentage of the total variance. One of the rules for the dimensionality reduction is to choose a number of eigenvectors which account for most of the variation of the eigenspectrum. Jolliffe [101] places this number approximately within the range of 70% to 90% of total variance. Based on the latter requirement, we are constrained to analyzing approximately the first $L = 290$ dimensions. Alternatively, applying a more subjective rule of "elbow" [101], this number reduces to $L = 8$ and $L = 30$. The extent of the flat part of the eigenspectrum ranges approximately from 70% to 90%. This is not unexpected in view of the fact that these numbers directly depend on the amount of noise-related line segments, $Nn = 302$ and precisely on the ratio $(100 - Nn/Ns) \cdot 100\%$. The deviation from Euclideaness is generally low for all four cases and ranges between $\epsilon = 0.8\%$ and $\epsilon = 4.3\%$.

To investigate the arrangement of reconstructed data points, we consider the first $L = 2$ dimensions and plot the first two eigenvectors $\mathbf{U}_1$ and $\mathbf{U}_2$. This leads us to the second important observation that highly distinctive structures do emerge within the feature space. Results presented in the first column of Fig. 4.7.2, illustrate the case of two groups of parallel line segments with equal P and D. The two-dimensional feature space shows a 90° corner with signal points arranged on each side. On the other hand, the "closed" structures studied in Fig. 4.7.2 (column 2) and Fig. 4.7.3 result in trajectory-like signal point arrangements. Regardless of the signal structures used in this synthetic experiment (shown in Fig.4.7.2 and Fig. 4.7.3), the plots of the first two corresponding eigenvectors exhibit a certain consistency in the constellation of feature space. Points which account for low-variance and non-intersecting structural noise are densely clustered near the origin, whereas the signal points occupy the outer feature space.

For insight into why the noise-related points appear densely clustered near the origin, it is not sufficient to consider the scale factor of the noise alone. It is rather the combination of the latter effect and of additional factors such as: (i) a 2D-PCA plot is the projection of multiple dimensions onto the 1^{st} and 2^{nd} eigenvector; (ii) approximately 80% of all data points account for the noise which is (iii) spread over approximately $L = 290$ dimensions. Therefore, this observation can be explained by the joint effect of scale and number of noise-related line segments and by the number of noise-related dimensions. The k-means clustering algorithm does not assume Gaussianity of the

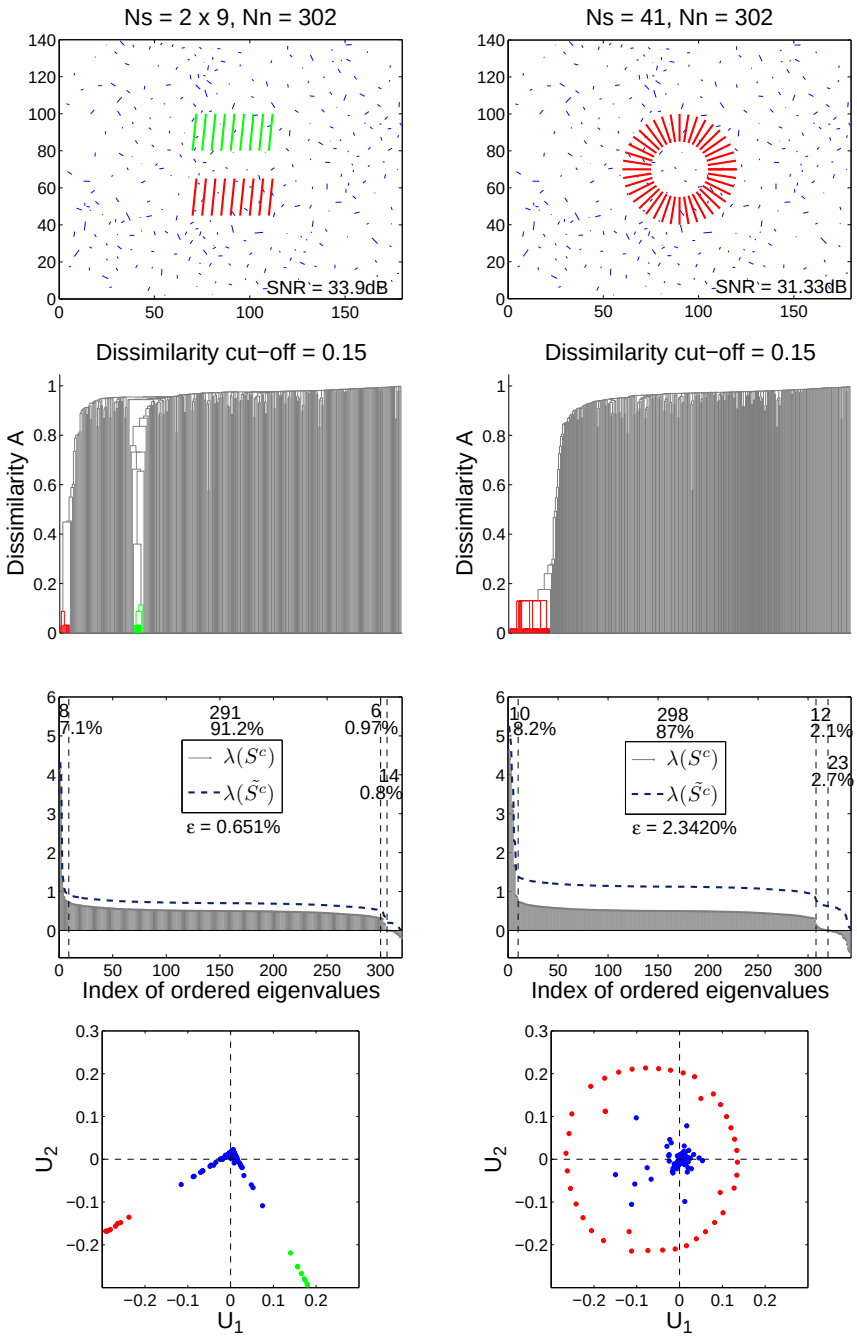


Figure 4.7.2: Synthetic structures with added stationary noise. Part I. **Row1**: Synthetic set of line segments with added noise and SNR values. **Row2**: Dendrograms constructed by using the symmetric dissimilarity matrix **A** and the single linkage algorithm. **Row3**: Plot of ordered eigenvalues of the pseudo-covariance matrix $\mathbf{S}^c$ and the CSE-embedded $\tilde{\mathbf{S}}^c$. The spectrum is partitioned into 4 bands, each showing the number of eigenvalues in that band and the percentage of total variance. **Row4**: 2-dimensional PCA space ($\mathbf{S}^c = \mathbf{U}\Lambda\mathbf{U}^T$, not CSE-embedded), showing the constellation of points (ground truth) corresponding to the line segments with structural signal (red, green) and noise (shown in blue) information. Diagrams are best viewed in color.

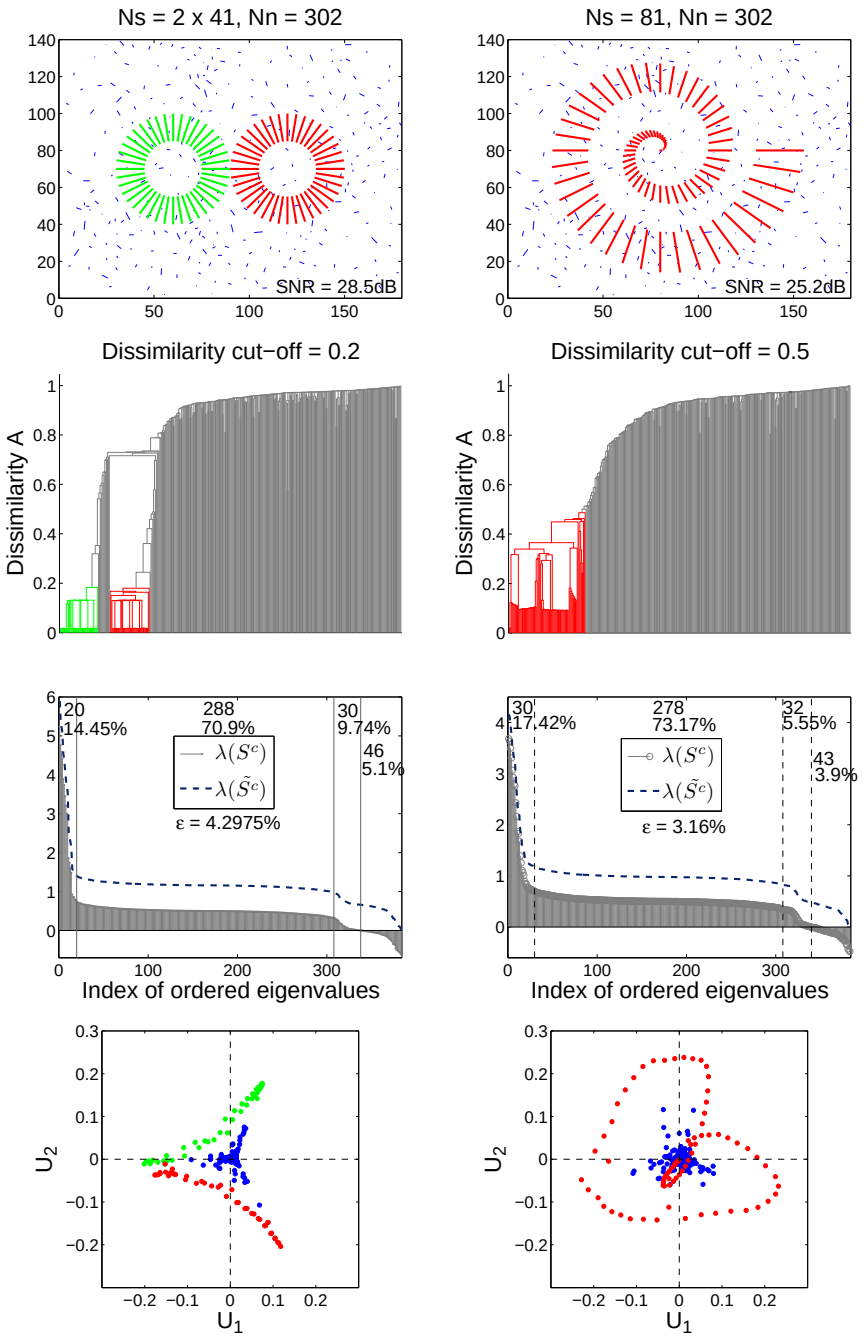


Figure 4.7.3: Synthetic structures with added stationary noise. Part II. **Row1**: Synthetic set of line segments with added noise and SNR values. **Row2**: Dendrograms constructed by using the symmetric dissimilarity matrix **A** and the single linkage algorithm. **Row3**: Plot of ordered eigenvalues of the pseudo-covariance matrix $\mathbf{S}^c$ and the CSE-embedded $\tilde{\mathbf{S}}^c$. The spectrum is partitioned into 4 bands, each showing the number of eigenvalues in that band and the percentage of total variance. **Row4**: 2-dimensional PCA space ($\mathbf{S}^c = \mathbf{U}\Lambda\mathbf{U}^T$, not CSE-embedded), showing the constellation of points (ground truth) corresponding to the line segments with structural signal (red, green) and noise (shown in blue) information. Diagrams are best viewed in color.

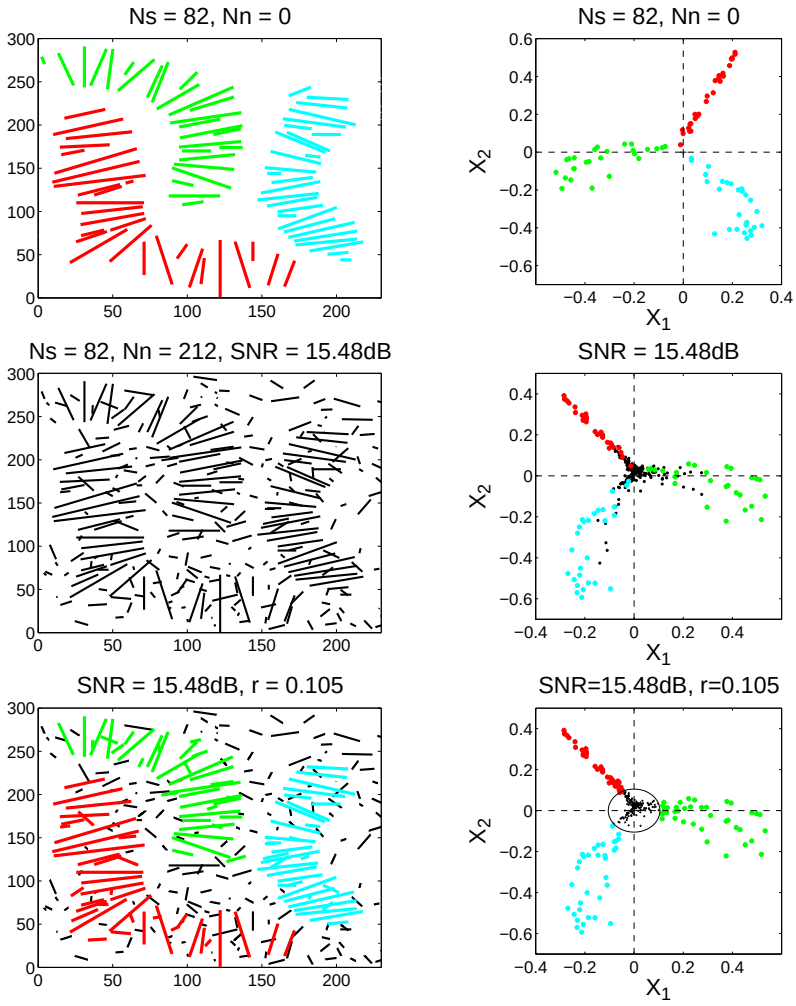


Figure 4.7.4: Synthetic line structures with added varying noise. Part I. **Row1**: Three synthetic signal objects with no added noise segments; **Row 2**: Addition of the structural noise-related line segments with $SNR = 15.48dB$. 2D plots of the reconstructed feature space show the arrangements of color-coded signal points and black-coded noise points. **Row3**: Grouping results in the image domain achieved with the noise pre-filtering with a circle of radius $r = 0.105$, and subsequent k-means with $k = 3$. Diagrams are best viewed in color.

data and works best with spherical or ellipsoidal clusters. These facts motivated us to include a de-noising stage prior to signal classification with k-means.

The third experiment on synthetic data sets involves three elongated signal structures and the addition of structural noise of increasing variance, as shown in Fig. 4.7.4 and Fig. 4.7.5.
The original arrangement (ground truth) of color-coded signal and noise points can be seen in the first row of Fig. 4.7.4. We observe that in the absence of noise, the three signal structures are represented by ray-like agglomeration of points, with the projected intersection point near the origin and with the relative orientation of approximately 120°. In a similar manner, the addition of $Nn = 212$ noise line segments and $SNR = 15.48dB$ resulted in the majority of noise-related data points clustered predominately near the origin. With decreasing SNR, noise-related line segments increase their interactions with signal-related line segments as the number and the amount of joint orthogonal projections in the image domain increases. In the reconstructed PCA feature space,

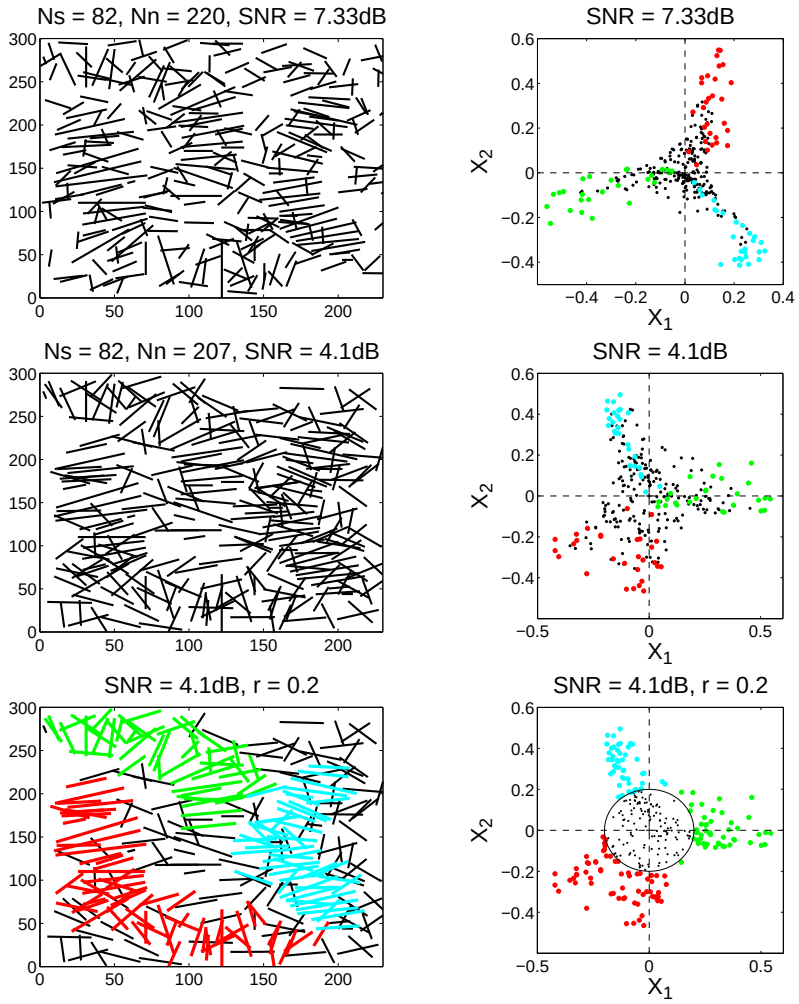


Figure 4.7.5: Synthetic structures with added varying noise. Part II. **Row1**: Experiment with added noise with $SNR = 7.33dB$; **Row2**: Experiment with $SNR = 4.1dB$. The 2D plot shows the actual ground truth signal and noise datapoints. **Row3**: 3-means grouping results. Noise filtering is implemented with a circle of radius $r = 0.2$. Diagrams are best viewed in color.

this translates to noise-related points spreading among the signal rays and progressively distorting the original constellation. For the purpose of clustering we apply our idea of de-noising the data starting outwards from the origin. This step is equivalent to defining a filtering circle with a radius r and performing the initial classification based on the circle membership. The third row in Fig. 4.7.4 shows the *3*-means clustering solution on the prefiltered data with a manually tuned $r = 0.105$, and illustrates how well the signal and the noise pattern can be differentiated by selecting the appropriate threshold. In other words, such a signal-noise classification results in a partitioning of the scene into foreground and background. In this context, the radius r may also be viewed as the *saliency* threshold of the scene. Although this formulation of saliency in a different domain agrees with the idea of Perona and Freeman [166], it differs in its definition.

The last experiment of $SNR = 4.1dB$, presented in the second row in Fig. 4.7.5, shows that signal and noise are almost inseparable by visual observation. Nevertheless, the three objects still can be detected by choosing the appropriated prefiltering radius of $r = 0.2$ and the subsequent 3-means clustering. This can be seen in the last row in Fig. 4.7.5. The patterns detected are contained

within the hull determined by the original signal structures. Generally, our synthetic experiments verify a more or less trivial fact that noise has a mitigating effect on the detection of original signal structures. Quantitatively, this is confirmed by the number of true positives we observe in the last row of Fig. 4.7.5. Surprisingly, under the experimental settings considered a noise appears to have a little effect on the hull detection. Quantitatively, this is evident in the number of false positives enclosed within the corresponding hulls. Further increase in r progressively shrinks the signal space until the points which correspond to the strongest, and thus the most salient R-links, remain. This fact agrees with our initial assumption that salient structures are represented by strongest R-similarity links. Furthermore, it suggests the applicability of the filtering approach to the task of mitochondria localization, where we primarily seek to determine the location of a subcellular structure, but not necessarily its exact composition.

In the last experiment with three synthetic mitochondria, we know in advance the number of k clusters. As can be seen in the first row in Fig. 4.7.4, for example, there are three non-convex signal structures in the feature space. In this case, the selection of $k = 3$ gives us the desired result as can be seen in the last row in Fig. 4.7.5. If we select either $k < 3$ or $k > 3$ then we do not obtain the correct partitioning. This is due to a number of facts, such as that the ground truth clusters are not convex, and that the addition of noise progressively decreases the inter-cluster variance while at the same time increases the intra-cluster variance in the feature space.

4.8 Application to Mitochondria Localization

In this section we describe experiments on subcellular TEM images of mitochondria. Grouping was performed on three sets of line segments extracted using line fitting operations on the edge map. In the first experiment which is illustrated in Fig. 4.8.1(Rows1-2), we examined grouping of line segments extracted from the TEM image of a *lamellar*-type mitochondrion from a mouse epididymal cell[4] In scatter plots of the reconstructed $\mathbf{X}$, shown in Fig. 4.8.1(Row 1), we observe two orthogonal ray-like clusters. According to (4.16), the low-dimensional feature space $\mathbf{X}_L$ has a dimension L. In the following, we define the distribution of the magnitude of $\mathbf{X}_L$ as:

$$\rho_L = \|\mathbf{X}_L\| \tag{4.18}$$

We use the distribution of $\rho_{L=2}$ to perform signal noise classification. The same formalism can easily be extended for $L > 2$. Ideally, we would prefer to have a well-separated bimodal distribution with one part representing the noise and the other representing the signal content. However, the first plot in the second row of Fig. 4.8.1 shows a multimodal and positively skewed distribution with the main density below mean. Our previous observation that the signal data tend to be concentrated in the outer range of $\mathbf{X}_L$ suggests that we need to consider the decaying tail of the ρ_2 distribution. In this example in Fig. 4.8.1(Row 2), we can see that the cut-off value of $r = \mu(\rho_2)$ yields two clusters which jointly represent the mitochondrial structure. The other possibility for obtaining the cut-off value worth considering, would be to decompose the multimodal distribution using a Gaussian Mixture Model. To address the question of how well our mitochondria detection works, Fig. 4.8.1(Row 2, last image) displays the localizations hulls on the original image.

[4]Image source: courtesy of ASCB (American Society for Cell Biology).

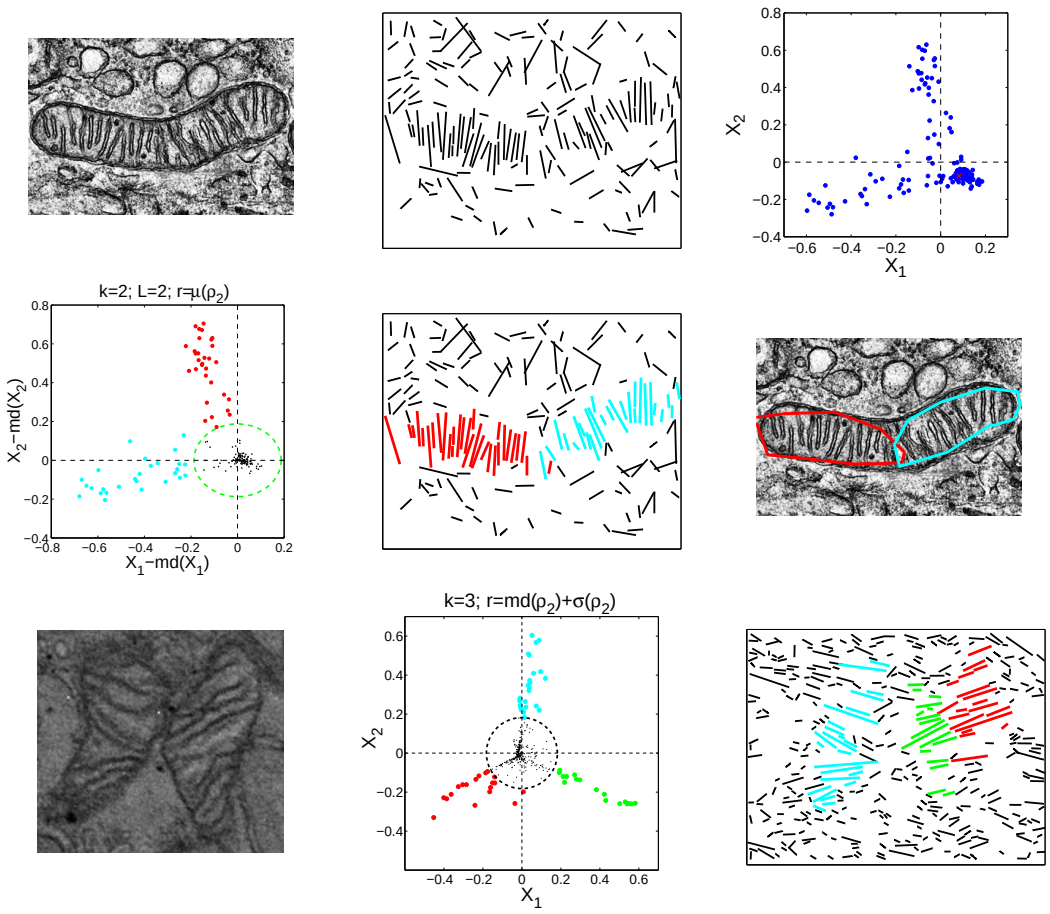


Figure 4.8.1: Application experiments. **Row 1**: TEM image of a *lamellar*-type mitochondrion from mouse epididymis; Extracted line segments; 2D-PCA of the reconstructed X data. **Row 2**: X data separation with $r = \mu(\rho_2)$ and the successive clustering of data outside the r-circle with $k = 2$; Clustering result in image domain; Localization hull superimposed on the original image. **Row 3**: TEM image of mitochondria from a human prostate cell treated with the apoptosis inducer STS; Three cluster solution to k-means clustering of 2D-X data outside the circle with the radius $r = md(\rho_2) + \sigma(\rho_2)$; Line clustering result in image domain. Diagrams are best viewed in color.

The second experiment was carried out on the TEM image of a human prostate cell[5], treated with apoptosis inducer staurosporine STS. The TEM image in Fig. 4.8.1 shows two adjacent mitochondria. The extracted kernel-PCA representation reveals three distinct ray clusters with a relative orientation of approximately 120°. In this example, the cut-off value for de-noising was chosen as $r = md(\rho_2) + \sigma(\rho_2)$, where $md(\rho_2)$ denotes the median of the ρ_2 distribution. The result of clustered line segments is shown in Fig. 4.8.1(Row 3, last image).

The third experiment was conducted on the TEM image of *tubular*-type mitochondria from hamster's adrenal cortex cell[6] as shown in Fig. 4.8.4(g). The set of extracted line segments which can be seen in Fig. 4.8.2(a), shows a weak presence of structural noise, yet six dominant signal structures. The distribution of points in the 2D-PCA space can be seen in Fig. 4.8.2(b). Based on our experience with synthetic elongated objects in Section 4.6, we anticipate the presence of

[5]Image source: RCSI (Royal College of Surgeons in Ireland).
[6]Image source: ASCB.

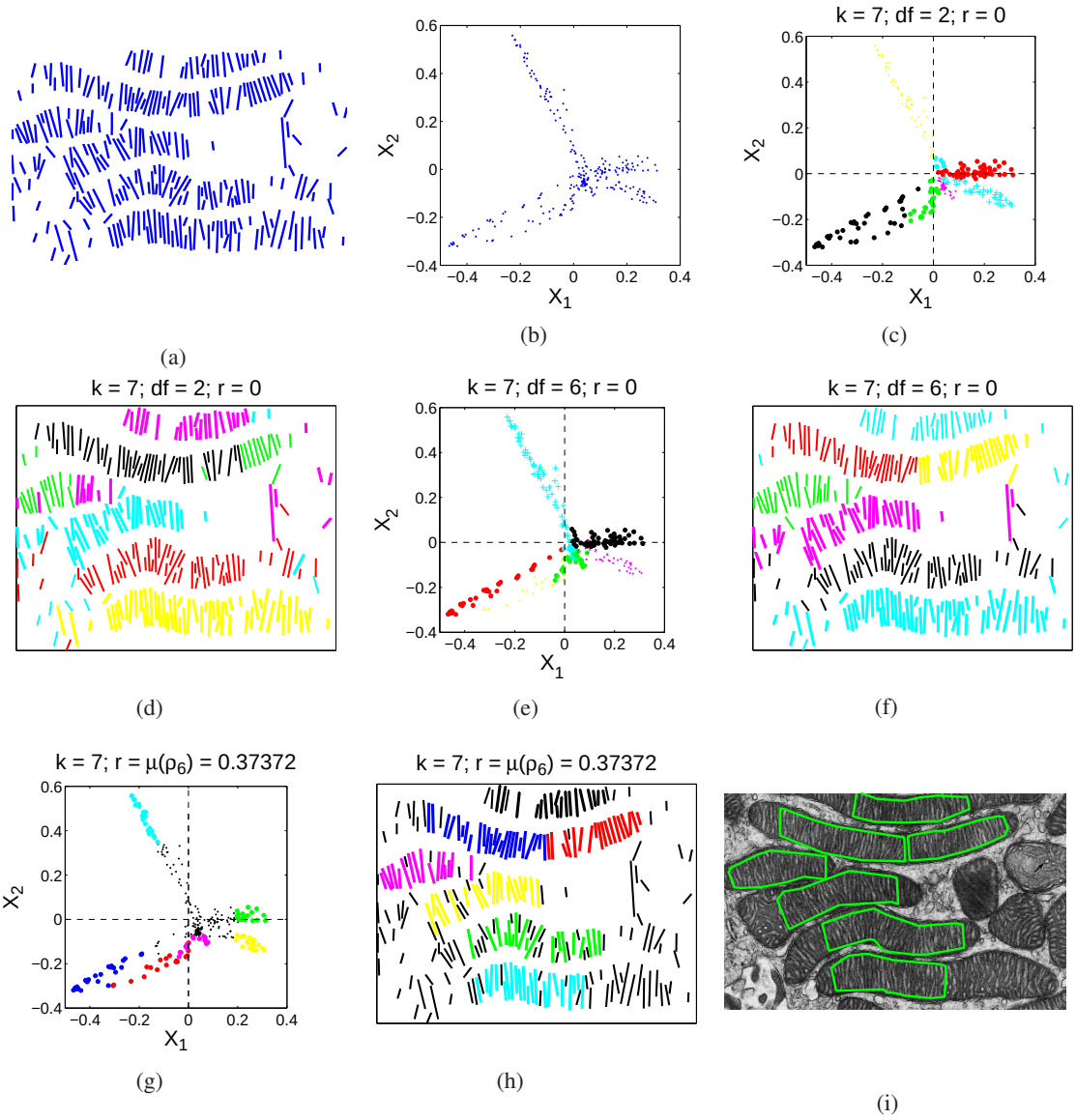


Figure 4.8.2: Clustering results for $k > 2$ and $L \geq 2$. **(a)** Extracted line segments: 6 visible clusters. We also see that the signal content is well above 50%. **(b)** 2D constellation of the reconstructed data $\mathbf{X}$: 4 visible ray-like clusters. **(c)** 7-means clustering ('Distance'='cosine') in $L = 2$ with $r = 0$. **(d)** Clustering results in image domain. **(e)** 7-means clustering in $L = 6$ with $r = 0$. **(f)** Clustering results in image domain. **(g)** 7-means clustering in $L = 6$ with $r = \mu(\rho_6)$. **(h)** Clustering results in image domain. **(i)** Localization hulls superimposed on the original image for the case of $L = 6$, $r = \mu(\rho_6)$, $k = 7$. Diagrams are best viewed in color.

six signal rays in the feature space. However, we only observe four major rays of data. Because of the different orientation of signal rays through the multiple dimensions, in the two-dimensional representation, the remaining two signal clusters may be scrambled within the four rays and be mixed with the noise-related data. This fact revokes the previous assumption that only the noise-related data can appear near the origin. It also suggests that more dimensions are needed to resolve the remaining two clusters.

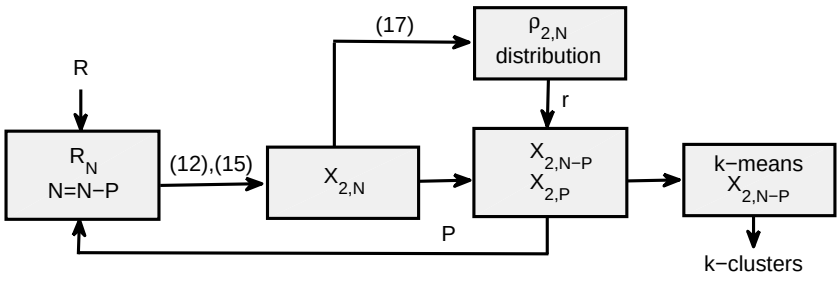


Figure 4.8.3: Proposed Dimensionality Unfolding Principle in L=2 dimensions.

To test this hypothesis, we experimented with finding a k-means solution in $L > 2$ dimensions. The results of this experiment are presented in Fig. 4.8.2. In order to obtain all seven (six signal plus one noise) clusters, we do not apply prefiltering in this experiment. Rather, we attempt to cluster the data according to the following methodology:

(i) in $L = 2$ dimensions by setting $k = 7$; see Fig. 4.8.2(c,d);
(ii) by increasing the number of dimensions to $L = 6$; see Fig. 4.8.2(e,f) and
(iii) by prefiltering the 6-dimensional data with $r = \mu(\rho_6)$ and the subsequent clustering with $k = 7$ and $L = 6$; see Fig. 4.8.2(g,h).

We conclude that the clustering results obtained in the image domain vary significantly for the above three cases. Because the basis for the unsupervised selection of parameters k and L is unclear, we propose an alternative *Dimensionality Unfolding* method which is illustrated in Fig. 4.8.3.

4.8.1 Dimensionality Unfolding Principle

We have seen that $L = 2$ dimensions are insufficient in detecting all six mitochondria. In this section, we propose a principle where clustering is repeatedly done in $L = 2$ dimensions and is performed after the partitioning of the data $\mathbf{X}$ in the i^{th} stage. We denote the partitioned data $\mathbf{X}^{(i)}$ for each clustering stage i. Partitioning depends on the amount of signal in the image and on how well the distribution of $\rho^{(i)}_{L=2}$ can be separated. In this example, as we can see from Fig. 4.8.4(h), the signal content (line segments describing mitochondria) is well above 50%. This fact justifies the initial $i = 1$ partitioning of $\mathbf{X}^{(1)}$ by taking the *second quartile* of the $\rho^{(1)}_2$ distribution (median point). The P points, which are located outside of the r-circle are sequentially grouped with the k-means algorithm. Next, we *unfold* the $N - P$ points inside the r-circle by computing the new kernel-PCA solution $\mathbf{X}^{(2)}$ on the *reduced* $\mathbf{R}_{N-P}$. In this stage, to extract the remaining unknown x% of the signal, we need to assume more than the 50% of the noise content. We can see in Fig. 4.8.4(e,f) that the remaining two clusters have emerged after the second $i = 2$ clustering stage with the cut-off value of $r = md(\rho^{(2)}_2) + \sigma(\rho^{(2)}_2)$. The combined localization hulls for all six mitochondria can be seen in Fig. 4.8.4(i). This localization result proves also to be the best achieved as compared to the $k > 2$ and $L \geq 2$ methodology, illustrated in Fig. 4.8.2.

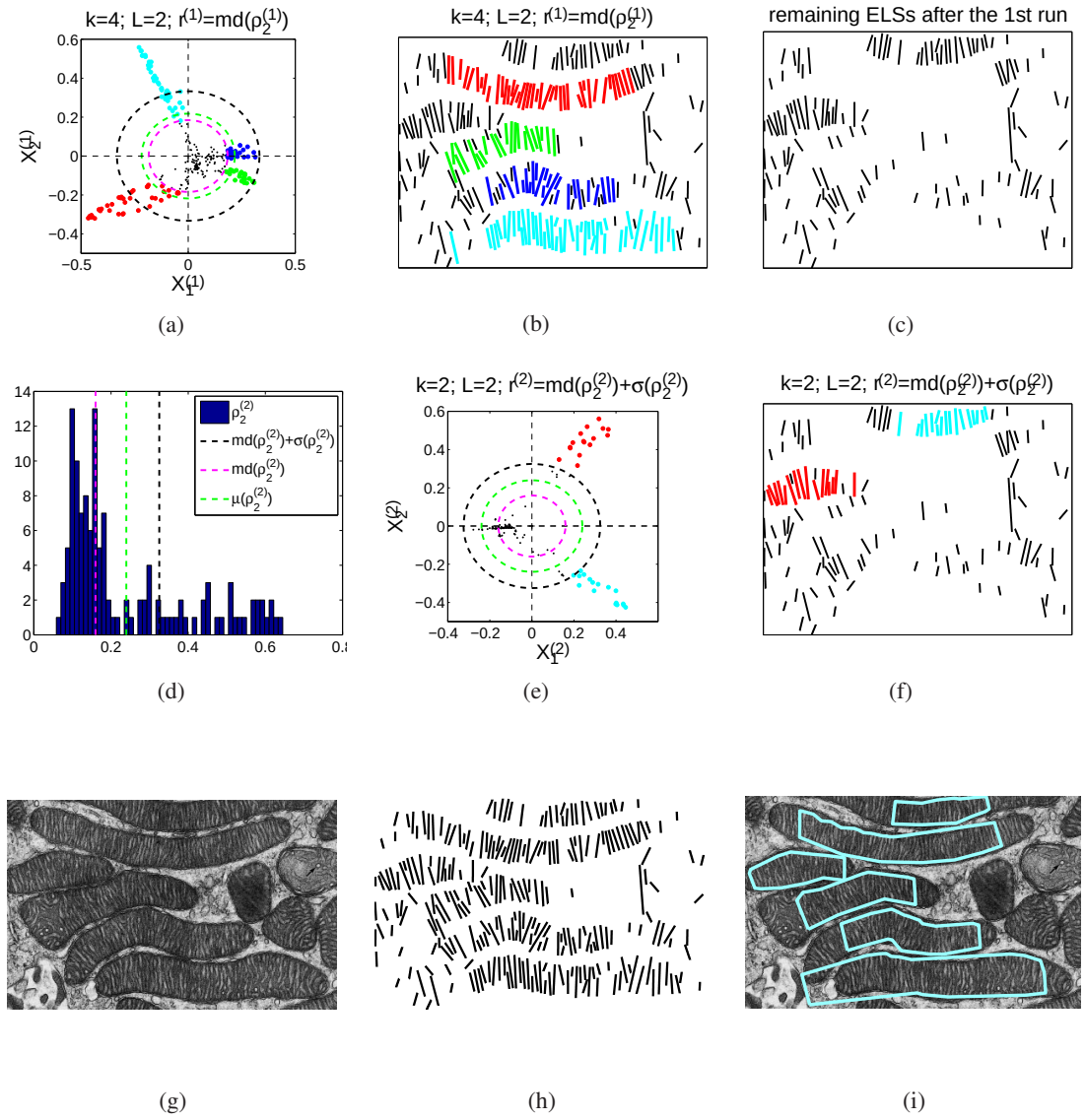


Figure 4.8.4: Results of the proposed Dimensionality Unfolding method in $L = 2$ dimensions. Application experiment on the TEM image of *tubular*-type mitochondria from hamster's adrenal cortex cell. **(a)** 4-means clustering solution for $\mathbf{X}^{(1)}$ outside the circle of radius $r^{(1)} = md(\rho_2^{(1)})$. **(b)** Image domain representation after the $i = 1$ clustering stage. **(c)** Remaining line segments which correspond to the points inside the r-circle. **(d)** $\rho_2^{(2)}$ distribution and separation thresholds. **(e)** 2-means clustering solution for $\mathbf{X}^{(2)}$ outside the circle of radius $r^{(2)} = md(\rho_2^{(2)}) + \sigma(\rho_2^{(2)})$. **(f)** Image domain representation after the $i = 2$ stage. **(g)** Original image. **(h)** Set of extracted line segments. **(i)** Final localization hulls as a combination of the intermediate results after $i = 1$ and $i = 2$ clustering stages. Diagrams are best viewed in color.

4.8.2 Results of Single Linkage Hierarchical Clustering

Single Linkage Hierarchical Clustering is the simplest and the oldest method of clustering which is robust to both non-Euclidean data and non-convex cluster shapes [9, 114]. It detects naturally occurring groups and allows the hierarchy of data to be examined. SLHC merges two clusters on the basis of the minimal distance between its entities, and can detect "U"-shaped and long serpentine clusters [9]. Though this chaining property, being a form of "*tracing*", is well suited for our

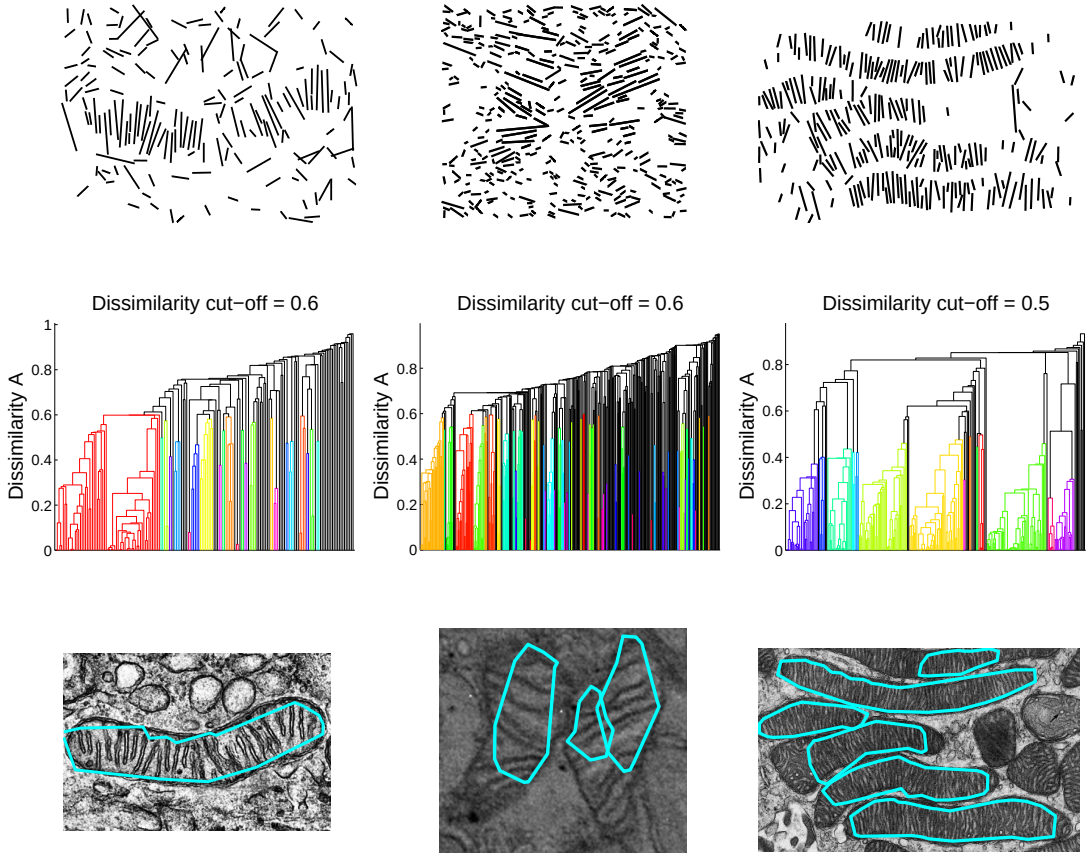


Figure 4.8.5: Single linkage hierarchical clustering results for mitochondria images. Localization is based on the analysis of the hierarchical cluster tree (SLHC). **Row 1**: Extracted line segments; **Row 2**: Dendrograms with manually selected cut-off values - resulting clusters are color-coded; **Row 3**: Localization hulls superimposed on original images. Diagrams are best viewed in color.

grouping objective, it is often considered to be a shortcoming in many other applications [9, 94]. The reason for this is that the two opposite ends of a cluster may be highly dissimilar or not be similar at all. Gibbons and Roth [78] remarked that in their application of clustering to gene expression analysis, the Single Linkage Hierarchical Clustering performs worse than random. In this section, we performed grouping directly on the *non-metric* dissimilarity matrix **A** by means of Single Linkage Hierarchical Clustering. The cut-off value, the size and the number of clusters were selected manually with the purpose of demonstrating the capability of the detection. The major shortcoming of Single Linkage Hierarchical Clustering, however, is the unsupervised decision on a suitable threshold for the analysis of the dendrogram. For the mitochondria segmentation application, we would first seek a localization marker inside a mitochondrion. Thus, a single pair of line segments describing one single inner membrane folding would suffice for that task. Therefore, we could start with a low threshold of dissimilarity values in the dendrogram, which would give us a high number of small clusters at first. Then, we could increase the threshold and evaluate the cluster properties and their compactness. However, too high a dendrogram cut-off value may produce incorrect partitioning results.

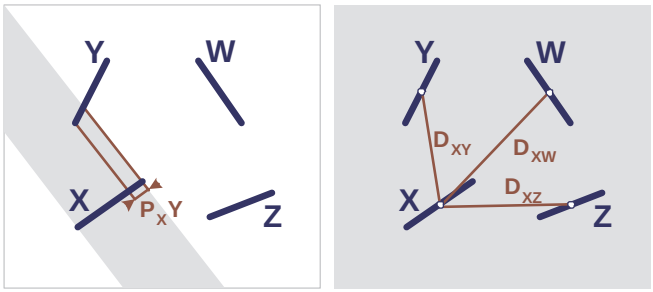


Figure 4.9.1: Illustration of the difference between restricted (a) and full (b) similarities for the case of $N = 4$ line segments X, Y, Z and W. **(a)** Left-sided orthogonal projections P onto the line segment X are defined by the number of line segments intersecting the plane generated by X. Only $P_X Y$ exists, while projections $P_X W = 0$ and $P_X Z = 0$. **(b)** Distance-based similarity D is defined for all line segments on the full space of $N = 4$. Diagrams are best viewed in color.

We realize by looking back at the "S"-shaped mitochondrion C shown in Fig. 4.3.1, that the tracing property of SLHC may be desirable in our case when applied to the localization of mitochondria of highly deformed shapes. The other important point to consider when grouping line segments of the mitochondrion C in Fig. 4.3.1 with a high length variability, is that the amount of orthogonal projections partly depends on the length of line segments. Therefore, it is possible that the kernel-PCA framework could benefit from normalization of raw data and grouping cues. Nonetheless, we observe and acknowledge that by choosing the appropriate dendrogram cut-off parameters, the SLHC-based localization results in Fig. 4.8.5 exceed those achieved in Section 4.8 which uses the kernel-PCA framework. On the other hand, the foreground to background segregation is incontestably well-defined within the PCA-based framework. Therefore one line of future research could consider the combination of both clustering methods: kernel-PCA to discard the noise-related line segments and to obtain the initial clusters of projected line segments, which would then be used as seed points for SLHC-based tracing.

4.9 Contour Integration with Rotated Line Segments

The critical question which needs to be answered is whether the signal-related pattern formation in the feature space is mediated by the intrinsic property of R similarity, or if it is conditioned by our experimental settings. To address this question we examine the existence rules of an orthogonal projection P of a line segment. Similarities which involve any pairwise combination of a distance or angle are defined for all N line segments (Fig. 4.9.1(b)), as naturally the distance and angle between all line segments simply exist. Conversely, the existence of orthogonal projections P onto a line segment X is restricted to a number much smaller than N, and it precisely depends on the number of line segments intersecting the orthogonal plane spanned by X (Fig. 4.9.1(a)). In our experiments we have seen that this fact leads to the initial classification of all non-projecting line segments as being not significant. We conclude that when combined with the proximity measure D in a ratio model, the restricted nature of orthogonal projections P aid the sparsity of the resulting similarity matrix $\mathbf{R}$ and promote the chaining effect of the strongest signal line segments.

Salient curves can also be viewed as being composed of *chained* line segments. This reasoning raises a question if the R similarity can potentially be used in the extraction of salient curves where traditional approaches primarily target the affine properties of line segments. Fig. 4.10.1(Row 1, left) shows a synthetic image similar to the one from Sha'ashua and Ullman's work on structural saliency in [185]. Fig. 4.10.1(Row 1, right) shows a synthetic image similar to the one from Perona and Freeman's work on affinity factorization in [166]. In order to describe local and global saliency, the authors considered similarity measures based on the co-linearity and the proximity of line segments. We have discussed this model in Section 3.4.1.2.

Generally, contour saliency relies upon the human ability to cognitively fill the gaps between adjacent line segments [24, 131, 166, 185] and also depends on the global context within which the contour appears [124]. Without any doubt, we detect two strongly salient contours in the first row of Fig. 4.10.1. By inspecting the second row of Fig. 4.10.1, we may also distinguish two structures, which in this case are composed of pairwise projected line segments. Based on these observations, we now consider a different view of the contour integration problem:

(i) in the original image we rotate each ELS by 90° around its center (see Fig. 4.10.1(Row 2))
(ii) in the transformed image we reinstate the problem of contour integration by using R.

This experimental setting also fundamentally differs from the one in Li and Gilbert's work [124], where the authors studied the effect of orientation on global contour saliency by rotating the entire stimuli rather than each particular line segment. With our methodology, the rotation of every line segment in the stimuli greatly disrupts the saliency, as can be seen in Fig. 4.10.1(Row 2). However, we still perceive the two signal structures as being significant. In this experiment we assume the rotational invariance of structural noise, generated under the conditions from Section 4.7. Fig. 4.10.1(Row 3) shows the now familiar segregation between the noise, which is clustered at the origin and the signal which occupies the outer region. With the saliency threshold $r = md(\rho_2) + \sigma(\rho_2)$, both contours have been successfully extracted, as demonstrated by the reversal of the rotation of all line segments illustrated in the last row of Fig. 4.10.1. These experimental results indicate that a *dual* relationship may exist connecting the R-based detection in 90° rotated space with the perceptual contour integration process in the original image.

4.10 Discussion

In this chapter we target the localization of mitochondria of lamellar and tubular morphology. We observe that in a densely cluttered scene, a unique criterion to localize a salient group of extracted line segments with different lengths, orientations and density may be stated as *Projectivity*. Further studies, however, are needed to support or reject the hypothesis that *Projectivity* has a link with perceptual organization. Furthermore, we introduce a new similarity measure which combines the *Projectivity* and the Proximity grouping principles. More precisely, it combines the amount of spanned orthogonal projections P and center-to-center distance D in a pair of line segments. We denote the new metric as *Projection-to-Distance* ratio.

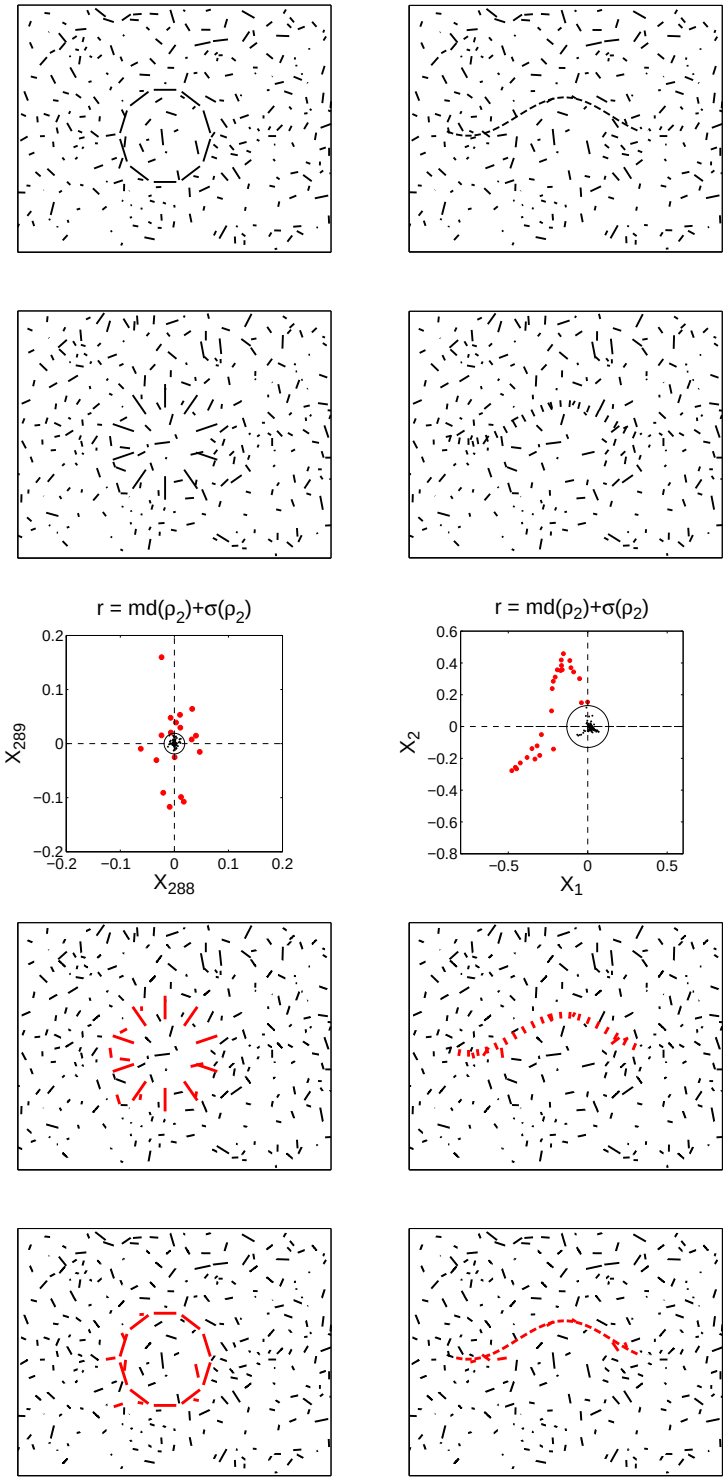


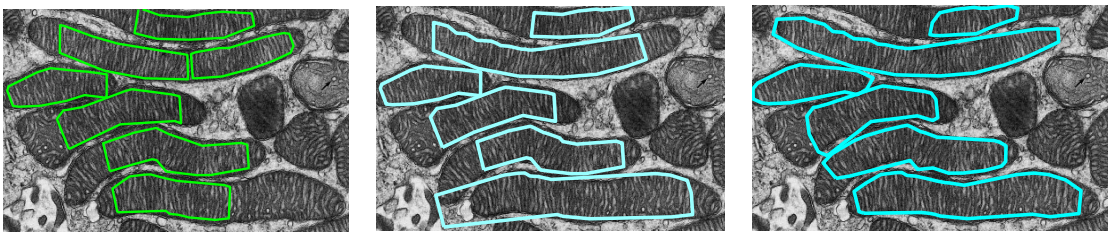
Figure 4.10.1: Contour integration experiment and proposed explanation of duality in the detection of salient contours. **Row 1**: Synthetic image similar to Sha'ashua and Ullman's work in [185]; Synthetic image similar to Perona and Freeman's work in [166]. **Row 2**: The same set of line segments as in Row 1, but all now are rotated by 90°. These data sets are used for the calculation of the *R* similarity. **Row 3**: 2D-PCA representation and foreground to background separation. **Row 4**: Clustering results in image domain. **Row 5**: −90° rotated line segments.

We conclude from the experiments conducted and described here that the most prominent feature of P is that it belongs to a class of restricted similarities and thus results in an inherently sparse similarity matrix. We apply our grouping approach to a set of line segments extracted from TEM images of mitochondria and observe that grouped line segments efficiently model the inner parts of organelles. While several experiments were conducted using mitochondria images, we would like to emphasize that the development of an automatic segmentation algorithm was not the main aim of this chapter. We rather addressed a theoretical problem concerning the role of projection similarity in line grouping, which has relevance to the extraction of salient features in cluttered backgrounds. We have shown that our method can be used to identify the hull of the salient structures in subcellular environments. However, the validation of our method would benefit from a larger database of mitochondria images and their ground truth segmentation for quantitative comparisons with other line grouping methods.

Investigations on the conformity of the introduced similarity measure with metric requirements reveals an inherently non-metric and thus a non-Euclidean behavior of the R similarity. We construct the similarity matrix $\mathbf{R}$ and analyze its properties, drawing the conclusion that it generally does not translate into the positive semidefinite pseudo-covariance matrix $\mathbf{S}^c$. We infer that precisely this fact constraints the use of available grouping methods and demands the strategy of intermediate Euclidean embedding. Furthermore, we apply Constant Shift Embedding and the kernel-PCA framework to the grouping of line segments in the reconstructed vectorial space. The experiments based on synthetic data sets yielded two main results. Firstly, we confirm that significant structures do emerge within the two-dimensional kernel-PCA space. Secondly, we observe that salient signal structures represented by the strongest R-links in the image domain manifest themselves as statistical outliers in the kernel-PCA feature space. Supported by these findings, we assume that the points densely concentrated near the origin can be interpreted as noise. Therefore, we first tried to implement a prior signal-noise classification by defining a prefiltering circle with a radius r. Our experimental results indicated that this step is equivalent to partitioning the scene in the image domain into foreground and background. Taken together, our results suggest that ρ_L as given in (4.18) can be associated with a *saliency* function.

We treat the detection of mitochondria not as a textural but rather as a purely structural problem. We have anticipated that the feature space would contain linearly separable clusters. However, we have also observed that for the large number of mitochondria to be detected, the two-dimensional kernel-PCA representation is not sufficient to resolve all mitochondrial structures, and that the selected kernel does not work in the way it should. Our results indicate that the success of optimizing the k-means solution in $L > 2$ dimensions is largely constrained by finding a tradeoff between the number of clusters k and the number of dimensions L. Based on this fact, we introduce the *Dimensionality Unfolding* principle. It can be viewed as a joint iterated partitioning and k-means clustering approach in $L = 2$ where each iteration involves *unfolding* of the data within the classification circle r and subsequent clustering with the k-means algorithm.

It is possible that the non-linearly separable feature space resulting from our selected grouping strategy is due to the fact that the combined kernel-PCA and k-means clustering framework is actually ill-posed for this problem and that "*tracing*" should be used instead. We did apply "*tracing*" alias nearest neighbor grouping of line segments by choosing single linkage hierarchical



(a) (b) (c)

Figure 4.10.2: Localization hulls superimposed on the original image of mitochondria. **(a)** Clustering results for $k > 2$ and $L \geq 2$. **(b)** Results achieved by using Dimensionality Unfolding principle. **(c)** Results of SLHC applied directly on the distance matrix **A**. Diagrams are best viewed in color.

clustering on the dissimilarity matrix **A** as has been shown in Section 4.7.2. Qualitative comparison of SLHC-based results for the tubular-type adjacent mitochondria is given in Fig. 4.10.2. This summary shows that Single Linkage Hierarchical Clustering (Fig. 4.10.2(c)) outperforms both the $\{k > 2, L > 2\}$ and the Dimensionality Unfolding grouping methodologies. The experiment shown in Fig. 4.10.2(a) returned small disjoint clusters which still could be used as a localization marker for the subsequent segmentation process. In comparison, Single Linkage Hierarchical Clustering-based localization (see Fig. 4.10.2(c)) resulted in the correct detection of all six adjacent organelles and in the best hull approximation of mitochondrial shapes.

The major difficulty encountered in this chapter, is with the distribution of points in the feature space. This fact lead us to the experiments with higher dimensions and number of clusters. It is also possible that by trying different kernels we can obtain more favorable and linearly separable clusters in the feature space. For example, the synthetic experiment in Fig. 4.7.1 showed the effect of using three different kernels on the geometry of clusters. Other remaining unsolved issues are the automatic selection of the number k of clusters in the k-means clustering, as well as automatic selection of the cut-off value for the analysis of dendrograms in SLHC.

It is clear that by using the symmetrization procedure on **R** and by retaining the symmetric $\mathbf{R}_{sym}$ while discarding the $\mathbf{R}_{skew}$, we are actually loosing some information contained in the asymmetric part. However, in order to make some inferences on if and how this fact may affect the feature point arrangement, we first need to quantify the degree of asymmetry, and also to benchmark against some clustering method which uses the whole asymmetric matrix **R**. These issues may be addressed in future work on line segments grouping. In Chapter 6, we propose a new anticorrelation-based formulation of spectral clustering which can then be directly applied on asymmetric pairwise proximity data. The dissimilarity data which originates from our proposed formalism of projected line segments can be seen as a contribution to the class of visually representative asymmetric and non-Euclidean data in a pattern recognition environment.

Chapter 5

Model-based Spectral Clustering Framework for Segmentation of Mitochondria

The unsupervised segmentation method proposed in the current study follows the evolutional ability of human vision to extrapolate significant structures in an image[1]. In this work we adopt the perceptual grouping strategy by selecting the spectral clustering framework, which is known to capture perceptual organization features, as well as by developing similarity models according to Gestaltic laws of visual segregation. Our proposed framework applies but is not limited to the detection of cells and organelles in microscopic images, and it attempts to provide an effective alternative to presently dominating manual segmentation and tissue classification practice. The main theoretical contribution of the work developed in this chapter resides in the formulation of robust similarity models which automatically adapt to the statistical structure of the biological domain and return optimal performance in pixel classification tasks under the wide variety of distributional assumptions.

5.1 Introduction

The most prominent challenge in human- and computer-based visual inspection of micrographs is the complexity and variety of biological structures at the micro- and nano-scales. The image characteristics of cellular and subcellular environments differ widely according to the organism, tissue type, anatomical sample preparation practices, staining protocols and not at least the applied imaging modality. A recent review on segmentation tools in electron tomography [12] emphasizes the need for more objective segmentation tools. We review the recent works on cellular and subcellular segmentation in Section 1.3 and Section 2.1. The conclusion on background research is that manual segmentation is still the present clinical practice.

A highly active research direction towards the pursuit of automation in biological image analysis is based on statistical machine learning within the class of *semi-supervised* methods [230].

[1]This chapter has been published in the Journal of Structural Biology (Elsevier, 2013) [50]

The guiding idea behind these approaches is that given manually labeled images, the goal is to learn models that can identify novel instances of the regions in test datasets [153]. Within this field, a commercial software package to support the automatic segmentation of subcellular structures such as mitochondria, for example, has yet to become available, and to date only a few applied solutions have been reported. Due to their central role in cellular bioenergetics, apoptosis and autophagy, the identification and classification of mitochondria is an area of increasing biomedical importance. A timely review of training-based pattern recognition techniques with application to biological image analysis is provided in [187], where the authors also emphasize the major limitation of the framework as being the considerable amount of data required during the training process.

Fully *unsupervised* learning methods, on the other hand, skip the training stage and do not necessarily require *a priori* knowledge about the structure of an object. Here, the partitioning of an image into a set of discrete clusters is implemented with dimensionality reduction techniques such as Principal Component Analysis, its non-linear extension to kernel-PCA, Linear Discriminative Analysis, Independent Component Analysis and many others [101]. In particular, spectral methods which consider eigenvector-based analysis of features [7, 54] are known to capture perceptual organization features, and to efficiently separate important (foreground) from non-important (background) information [7, 166]. This ability is based on the fact that the leading eigenvectors of a similarity matrix account for most variability in the data and contain the information about the most dissimilar regions of an image. A similarity model that serves as an input to a spectral clustering algorithm, encompasses various image features such as texture, color, motion, brightness or higher level primitives such as extracted line segments and contours.

In this work we utilize the perceptual grouping strategy, firstly by selecting the spectral clustering framework and secondly by developing our similarity models according to Gestaltic laws of visual segregation [24, 62], such as *Closure*, *Proximity*, *Similarity* and *Figure-Ground*. Within this framework, we view a mitochondrion as a closed membrane organism and specifically target the detection of membranes. Therefore, we approach the diversity challenge of mitochondrial morphologies by targeting the global appearance properties of membranes in electron microscopy images. One immediate asset of our approach is that the outer membrane is the only descriptive feature of a mitochondrion which can be considered invariant under the apoptotic (pathogenic) changes of inner mitochondrial morphology [199]. Another advantage is based on the fact that perceptual systems reflect scale-invariance of the environment provided that they are adapted to its statistical structure [34]. This inference is of special importance when working with large scale images, and allows us to apply low-resolution block-based processing with the objective of reducing computational load. Finally, the manually annotated data used in our study for algorithm validation builds an additional interface between artificial and human vision. Another interdisciplinary link with human vision and psychology is the clinical annotation of micrographs which is the result of human perception, experience and judgement towards the validity of the manual mitochondria segmentation performed under often challenging conditions.

The major contribution of our work is the development of an *unsupervised* segmentation framework based on a combination of principles governing perceptual organization and unsupervised machine learning. In Section 5.3 we propose the concept of adaptive similarity mixtures which eliminates the need for user intervention in the selection of parameters.

Table 5.1.1: Notations used throughout this chapter.

n	number of pixels in an image
$\mathbf{S}$	similarity matrix
$\mathbf{A}$	kernel matrix (dissimilarity matrix)
$\mathbf{C}$	double-centered kernel matrix
$\mathbf{1}_n$	n vector of ones
$\mathbf{I}$	$n \times n$ identity matrix
$\mathbf{Q}$	projection matrix on the orthogonal complement of $\mathbf{1}_n$
$\mathbf{U}$	matrix containing eigenvectors
$\mathbf{\Lambda}$	matrix containing eigenvalues on the main diagonal
$\mathbf{\Phi}$	kernel-PCA based feature vectors
L	number of dimensions
k	number of k-means clusters
c	image downscaling factor
f	Gaussian filter size
G	Gaussian-filtered image
$\bar{G}$	inverted Gaussian-filtered image (membranes are in bright pixels)
∇	image gradient
I	image intensity
F	intensity-based similarity nominator
D	distance-based dissimilarity denominator
$\mathbf{Z}$	Z-score based matrix
α, β, γ	adaptive similarity parameters (5.5)

In the experimental Section 5.4, we validate our algorithm on two datasets of mitochondria[2]. These datasets have been selected to test the sensitivity of detection, and exhibit such unfavorable image characteristics as low contrast, non-uniform illumination, speckle noise, absence of staining markers, varying image sizes and varying resolution.

5.2 Recursive Spectral Clustering Framework

In order to compare n image features we construct the corresponding $n \times n$ similarity or *affinity* matrix $\mathbf{S}$ and the corresponding dissimilarity matrix $\mathbf{A}$. Although the spectral methods differ widely in the type of normalization of the matrix used for diagonalization, in this chapter we adopt the clustering framework based on kernel-PCA, Euclidean embedding and k-means that is detailed in [176]. The option of non-linear *kernelization* of the feature space allows us to employ a clustering algorithm with the linear cut capability, such as k-means. We briefly review the main processing path of kernel-PCA and introduce the notion of a centralized dot product matrix $\mathbf{C} = -0.5\mathbf{QAQ}$, with $\mathbf{Q} = \mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$. Here, $\mathbf{I}_n$ denotes an $n \times n$ identity matrix, $\mathbf{1}_n = (1, 1, ..., 1)^\top$ is an n vector of ones and $\mathbf{Q}$ is the projection matrix on the orthogonal complement of $\mathbf{1}_n$. Reconstructed data $\mathbf{\Phi}$ are obtained through spectral decomposition of $\mathbf{C} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ and the subsequent re-scaling by applying $\mathbf{\Phi} = \mathbf{U}\sqrt{\mathbf{\Lambda}}$ [176].

[2]Provided by the ASCB (American Society for Cell Biology).

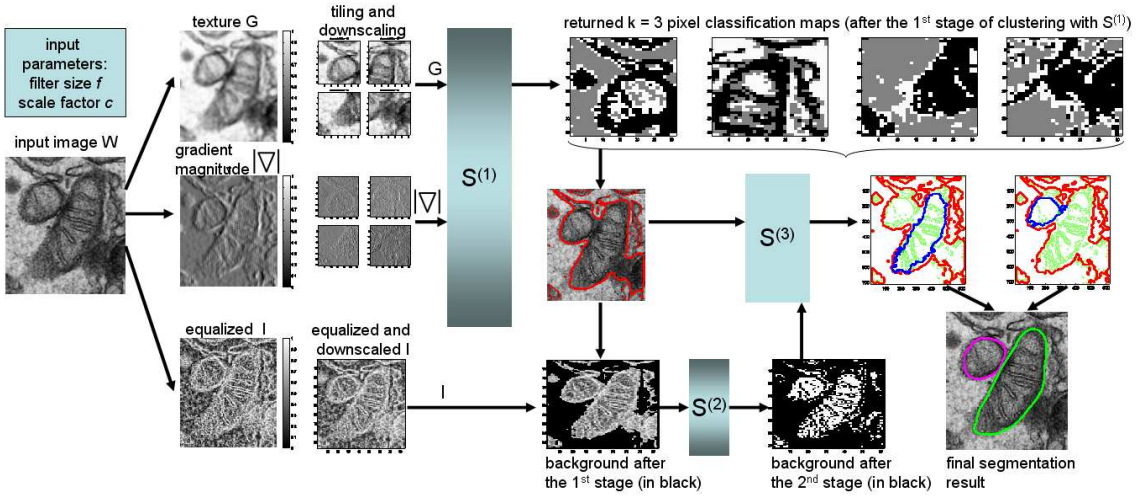


Figure 5.2.1: Proposed processing pipeline showing the progression from the input electron microscopy (EM) image on the far left to the set of detected organelles in the far right images. The three clustering stages $m = \{1, 2, 3\}$ are denoted by $S^{(m)}$. Diagrams are best viewed in color

The rows of $\mathbf{\Phi}$ correspond to the multidimensional vectors $\varphi_i, i = 1, ..., n$. The dimensionality reduction step is carried out by taking the first L columns of $\mathbf{\Phi}_L = \mathbf{U}_L \sqrt{\Lambda_L}$. We retain the leading component Φ_1 and apply k-means in the feature space to obtain the set of discrete clusters.

The complete processing path is shown in Fig. 5.2.1 and does not include any *a priori* models for mitochondria shapes, textures, or the customized energy functionals which are usually needed for training purposes. The main hypothesis employed in the development of our segmentation solution was based on the observation that subcellular organelles are defined by closed structures with distinct inner textures. Thus, we attempted to exploit specific knowledge in relation to the morphology and structural characteristics of mitochondria and to target the global appearance properties of mitochondrial membranes. For that purpose, our designed similarity models specifically integrate the intensity and spatial properties of the mitochondria membranes.

The segmentation architecture proposed consists of a multi-stage largely unsupervised scheme in which the first step involves data preprocessing in order to reduce noise and to enhance local structure coherence. In the proposed architecture we segment the membranes first, then fill the inner part of the mitochondria and separate the inner and outer wall of the outer membrane. Finally, we apply contour matching to segment adjacent organelles and to remove other connected blob-like or membrane-line objects which do not belong to mitochondria. Due to the size of the high-resolution images, we designed our algorithm to work with tiled and further downscaled (by a factor of c) images. Then, from the constructed similarity matrix for each tile, the global salient and non-salient membrane structures can be determined by means of spectral clustering.

In the first stage of clustering our objective is to close and segment dark membranes. For the purpose of membrane closure, we create a filtered image G by convolving the original input image with a designed Gaussian-based filter, multiplying the result with the original image and further re-normalizing the final product to the $[0, 1]$ range. The degree of *closure* is controlled by the scale of the Gaussian filter. The gradient magnitude $|\nabla|$ is also used in the similarity model $S^{(1)}$ in the first clustering stage. However, the gradients obtained are weak as they are caused by low

intensity profiles that are often characteristic for mitochondria membranes. Separate classification maps for each tile are further combined to yield the binary result of figure-ground segmentation of membranes. A postprocessing step based on mathematical morphology performs the flood-fill of the inside of mitochondria. The final outer contour (shown in red) can be seen in Fig. 5.2.1. In the second recursive clustering stage, we recluster the filled foreground obtained, with a different similarity model $S^{(2)}$, where the objective is to classify the inner part of a mitochondrion. By doing so we aim to separate the inner membrane from the outer membrane, and to obtain the inner part of a mitochondrion as a convex contour where the inner folding is included in that contour. The third stage of clustering $S^{(3)}$ involves a contour processing step and the computation of a minimum distance between the set of extracted inner contours $S^{(2)}$ and the outer contour $S^{(1)}$. The final segmentation result in Fig. 5.2.1 shows the identification of adjacent organelles.

5.3 Proposed Adaptive Similarity Models

In this work we connect the notion of adaptivity with non-parametric learning models [24, p. 32] which can adjust to the biological domain without user intervention and perform equally well under a wide variety of distributional assumptions. In many studies, however, the selection of parameters to accompany a similarity model is a sensitive issue and is primarily data-driven [25, 166]. One example of a state-of-the-art similarity model is given in (5.1) [25, 166] and involves squared difference of intensities I between two pixels p_1 and p_2:

$$F_{\Delta I}(p_1, p_2) = \exp[-(I(p_1) - I(p_2))^2/a] \tag{5.1}$$

where a is a penalization parameter. The goal of (5.1) is to discover regions with small variations in intensities. Thus small intensity differences ΔI representing similarly bright pixels are weighted equally with similarly dark pixels of the same ΔI. Such a model, therefore, is inherently inefficient in capturing structures represented by bright pixels. This limitation is of particular importance to the selected perceptual grouping strategy, as in a complex visual arrangement of gray and white pixels, the human attention is easily drawn to brighter areas – a phenomenon based on the Gestalt law of *Figure-Ground* [62].

In this work and in the first clustering stage, we introduce an additional intensity-based term $F_1^{(1)}$ which weights similarly bright pixels higher than similarly dark pixels in the inverted filtered image $\bar{G}$ where the salient membrane features are given by bright pixels. In order to weigh the pixels occurring near to the gradient edge higher, we utilize in addition the inverse exponentially mapped and normalized gradient magnitude $|\nabla|$ of the input image. Our final intensity-based function $F^{(1)}$ is a maximization between the brightness criterion represented by $F_1^{(1)}$ and the region-based contrast criterion represented by $F_2^{(1)}$. The weighting constants k_1 and k_2 provide an option to balance the influence of an either term, as shown in (5.2):

$$\begin{aligned}
F_1^{(1)}(p_1, p_2) &= k_1 \cdot [\bar{G}(p_1) \cdot e^{-|\nabla|(p_1)} + \bar{G}(p_2) \cdot e^{-|\nabla|(p_2)}], \\
F_2^{(1)}(p_1, p_2) &= k_2 \cdot [\max(\bar{G} \cdot e^{-|\nabla|}) - |\bar{G}(p_1) \cdot e^{-|\nabla|(p_1)} - \bar{G}(p_2) \cdot e^{-|\nabla|(p_2)}|], \\
F^{(1)}(p_1, p_2) &= \max(F_1^{(1)}, F_2^{(1)}), \\
F^{(2)}(p_1, p_2) &= \exp[-|I(p_1) - I(p_2)|] \, .
\end{aligned} \tag{5.2}$$

We use the gradient magnitude-based term to include sharpness variation in microscopic images. The term included in square brackets in $F_2^{(1)}$ is the normalized region-based contrast criterion, and the exponential mapping is included for the same reason of normalization. $F^{(1)}$ is the winning choice between the brightness and contrast criteria. The last equation in (5.2) shows the intensity-based model $F^{(2)}$ for the second recursive stage of spectral clustering where the objective is set to recluster the foreground extracted after $S^{(1)}$. In order to include the Gestalt law of *Proximity* into our model, we consider the squared Euclidean distance D between pixels p_1 and p_2. In the final similarity model, our objective is to let F decay with increasing D in a ratio relationship. In order to alleviate the problem that coordinates and intensity are measured on different scales and in different quantities, we propose the following three normalization procedures (see also Fig.5.3.2).

First, contrary to formulations presented for example in [25, 166], we avoid user-based selection of a distance penalization parameter σ_D^2 by defining D in terms of Z-score normalized coordinates z_x and z_y:

$$D^{(1)}(p_1, p_2) = D^{(2)}(p_1, p_2) = \exp[(z_{x1} - z_{x2})^2 + (z_{y1} - z_{y2})^2] \tag{5.3}$$

Second, in every clustering stage we construct two matrices $\mathbf{F}$ and $\mathbf{D}$ according to pairwise similarities (5.2) and pairwise distances (5.3), and define the final similarity matrix $\mathbf{S}$ as a pairwise ratio of $\mathbf{F}$- and $\mathbf{D}$-based Z-scores, denoted $\mathbf{Z}_F$ and $\mathbf{Z}_D$, where higher Z-scores correspond to higher probability of non-accidental occurrence. Standardization is attained with arithmetic mean μ and standard deviation σ, both being computed over all matrix entries [44].

Third, in order to control the degree of mixing $\mathbf{Z}_F$ and $\mathbf{Z}_D$, we introduce two constants, $\alpha \in \mathbb{R}^+$ and $\beta \in \mathbb{R}$, in the denominator, as can be seen in (5.5). In this formulation, the parameter α can be thought of as the standard deviation of the new distribution:

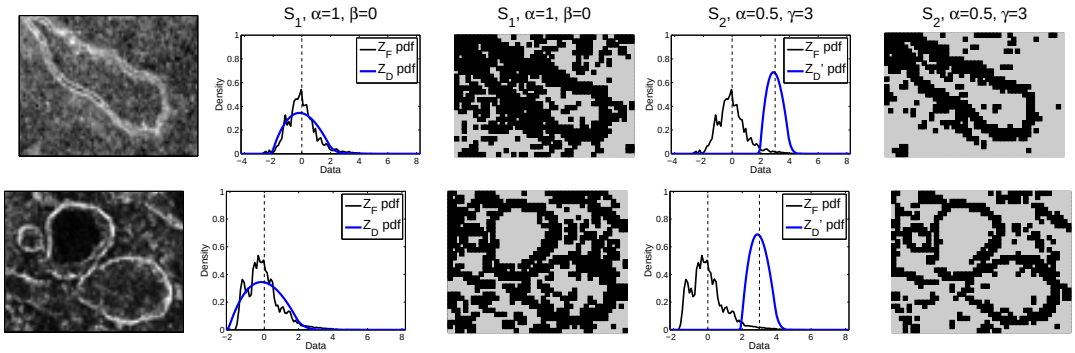
$$\mathbf{Z}'_D = \alpha \cdot (\mathbf{Z}_D + \beta), \tag{5.4}$$

and the parameter β can be thought of as the mean of $\mathbf{Z}'_D$. Thus, α controls the increase or decrease in the variability of the $\mathbf{Z}'_D$ distribution, and β controls the shift of $\mathbf{Z}'_D$ to the left for $\beta < 0$ or to the right for $\beta > 0$, where the goal is to align the right tail of $\mathbf{Z}_F$ and the left tail of $\mathbf{Z}_D$. In every clustering stage m, the final similarity matrix $\mathbf{S}^{(m)}$ is obtained according to:

$$\begin{aligned}
&\mathbf{S}^{(m)} : s_{ij}^{(m)} = z_{Fij}^{(m)} / [\alpha^{(m)} \cdot (z_{Dij}^{(m)} + \beta^{(m)})], \\
&\beta^{(m)} = \gamma^{(m)} + |min[\mathbf{Z}_D^{(m)}]|, \\
&\gamma_{opt}^{(m)} = max(\mathbf{Z}_F^{(m)}), \\
&\alpha_{opt}^{(m)} = 1 \ .
\end{aligned} \tag{5.5}$$

For the case of an approximately *Gaussian* probability density function of $\mathbf{Z}_F$, one possible scenario for optimal sampling is to position the left end of $\mathbf{Z}_D$ at $3\sigma_{ZF} = 3$, which captures 99.7% of the random variation in $\mathbf{Z}_F$. Applications seeking to increase the influence of the distance term may scale $\mathbf{Z}_D$ by $\alpha < 1$.

For the general case of a *non-Gaussian* pdf of $\mathbf{Z}_F$, we propose to align the right end of $\mathbf{Z}_F$ with the left end of $\mathbf{Z}_D$ (5.5), and by doing so to completely separate the two distributions.



(a) (b) (c) (d) (e)

Figure 5.3.1: Concept of adaptive similarity mixtures $\mathbf{S}$ where $\mathbf{F}$ and $\mathbf{D}$ are modeled as probability density functions $\mathbf{Z}_F$ and $\mathbf{Z}_D$ through the introduced standardization of matrix elements. The bright membrane structures (here in inverted EM images) are captured in the right half of $\mathbf{Z}_F$

Two examples in Fig. 5.3.1 are provided to illustrate that by varying α and γ we can control the accuracy of membrane detection in terms of thickness and gap completion. Specifically, by comparing columns (c) and (e) in Fig. 5.3.1, it can be seen that moving $\mathbf{Z}_D$ away from $\mathbf{Z}_F$ (compare Fig. 5.3.1(b) and Fig. 5.3.1(d)) reduces the amount of false positives identified.

The final step in data preparation before proceeding with spectral decomposition is the dissimilarity mapping and kernelization. These procedures are necessary in order to transform the final similarity matrix $\mathbf{S}$ into a centralized pseudo-covariance matrix $\mathbf{C}$ which is used later as an input to kernel-PCA (explained for example in [176]). For our two spectral clustering stages we applied the linear dissimilarity mapping, and for the first stage $\mathbf{C}^{(1)}$, we additionally applied the square root kernel:

$$\begin{aligned} \mathbf{C}^{(1)} &= -\tfrac{1}{2}\mathbf{Q}\sqrt{(\max(\mathbf{S})-\mathbf{S})}\mathbf{Q}, L^{(1)}=1, \quad k^{(1)}=3 \\ \mathbf{C}^{(2)} &= -\tfrac{1}{2}\mathbf{Q}(\max(\mathbf{S})-\mathbf{S})\mathbf{Q}, L^{(2)}=1, \quad k^{(2)}=2 \end{aligned} \tag{5.6}$$

After the diagonalization of $\mathbf{C}$, the final set of discrete clusters is obtained with a k-means clustering algorithm on the retained first ($L=1$) kernel-PCA output feature vector Φ_1.

As can be seen in (5.6), the first clustering stage returns 3 clusters $\{k_1, k_2, k_3\}$, where the decision on the foreground is made in the image domain according to $\mu(W(k_1)) < \mu(W(k_2)) < \mu(W(k_3))$. Thus we select the cluster with the minimal average brightness. Our choice of the 3-class partition was motivated by the assumption that with $k > 2$, we can segment thinner structures while closing the membrane gap with the width d and at the same time maintaining the separation of close lying organelles with the same d. The above mentioned effect of $k = 3$ can, for example, be observed in the returned pixel classification maps in Fig. 5.2.1.

The third clustering stage $S^{(3)}$ is outside the spectral clustering framework. For the extracted inner contours obtained after $S^{(2)}$ we compute and evaluate Euclidean distances between the inner contour and the outer contour obtained after $S^{(1)}$ with the objective of retaining only those outer contour points with the minimal distance to the currently analyzed inner contour. Our proposed four-stage cascaded raw data normalization and data fusion is illustrated in Fig. 5.3.2. This diagram shows how the unnormalized input data are factored into the joint similarity model S.

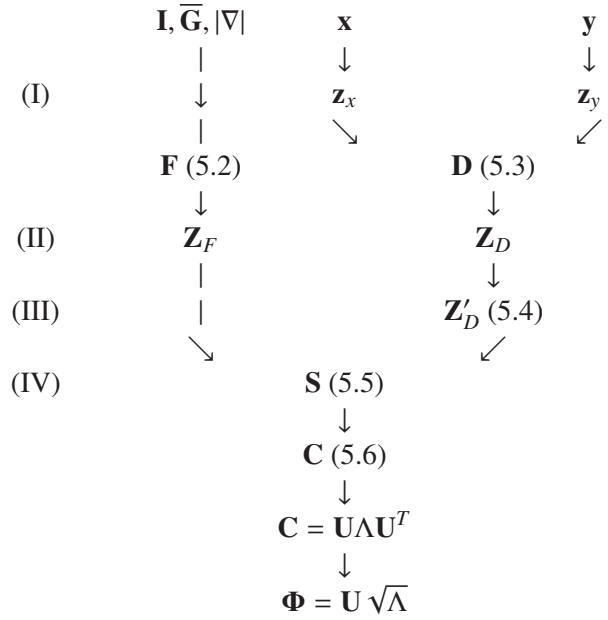


Figure 5.3.2: Proposed four-stage cascaded raw data normalization and data fusion. Stage (I) consists of standardization of the raw coordinates x and y. Stage (II) consists of the computation of the Z-score matrices $\mathbf{Z}_F$ and $\mathbf{Z}_D$. Stage (III) determines the optimal shift of $\mathbf{Z}_D$. Stage (IV) determines the final data fusion.

5.4 Experimental Results

We applied our model to two datasets each consisting of 17 EM images provided by the American Society of Cell Biology. The images contain mitochondria from the *ductulus efferens* of the ground squirrel, *Cittelus lateralis* [65]. All mitochondria of this type have an elongated profile. However, some organelles appear spherical or elliptical with the "washed-out" cristae and/or outer membrane and some organelles contain shadow characteristics in the outer region as can be seen in Fig. 5.4.2(a). These problems are related to the section thickness in typical EM studies ranging from 50-90nm and thus not being thick enough to include the whole mitochondria [65]. Thus, the shadow artefacts mentioned above are the result of tangentially oriented membrane within the section thickness. Therefore, based on the suggestion made by our biomedical partners, the data annotation was carried out as a smooth closed contour extrapolation. Figure 5.4.1 shows two examples of manual annotation of the inner and outer mitochondrial membranes.

In order to construct the Dataset I, we extracted the regions of interest at different magnifications. Thus, images in the first dataset have different resolution and therefore the thickness of the membrane is assumed to be unknown. In the acquisition of the second dataset, we applied equal magnification while extracting the regions of interest. All images have different sizes and generally the image quality and the contrast are extremely low, which on the other hand allows us to test the sensitivity of our detection algorithm. The weighting constants k_1 and k_2, introduced in (5.2), are selected according to the robust Qn spread estimator of the input data G [44] such that $k_1 = \sigma^2_{Qn}(G)$ and $k_2 = 1 - \sigma^2_{Qn}(G)$.

Selected qualitative segmentation results for the ASCB Dataset I are provided in Fig. 5.4.2. Here, the comparison is given between the original image, manually annotated and further clini-

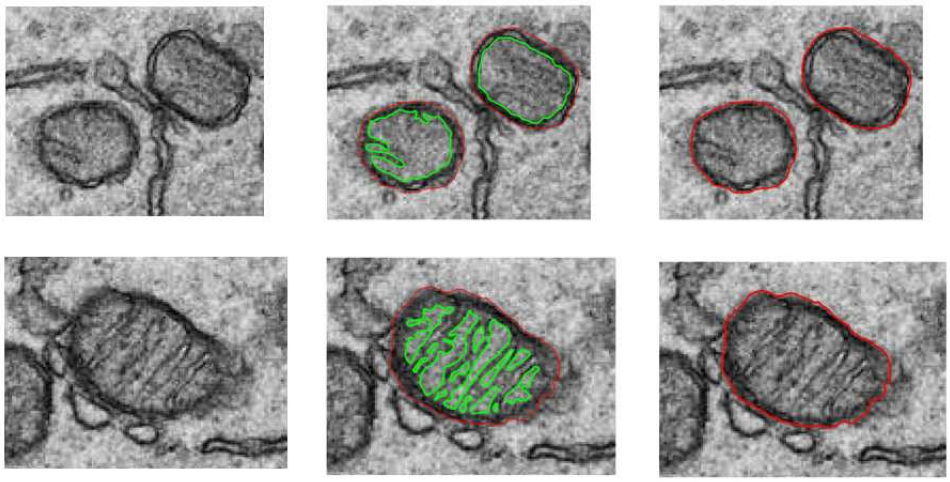


Figure 5.4.1: Ground truth manual segmentation with Adobe Photoshop 7.0. All annotations have been validated by our biomedical partners from RCSI. Diagrams are best viewed in color.

cally validated ground truth data, our segmentation results after the first clustering stage $S^{(1)}$ and our final segmentation results of the outer mitochondrial membrane. Selected segmentation results on the ASCB Dataset II are provided in Fig. 5.4.3.

The 2D user-interactive semi-automated tracing software *Livewire* is a popular medical segmentation tool [200]. Fig. 5.4.4 provides the results of the segmentation of mitochondrial outer membrane, achieved by using *Livewire*. Specifically, these segmentation results demonstrate high accuracy in semi-automated segmentation using this tool as compared to the ground truth data of these organelles in Fig. 5.4.2 column (f).

For quantitative assessment, the result of segmentation and the ground truth data is given by *filled* closed segmented contours, such that the foreground object (mitochondria) is 1 (white) and the background is 0 (black). In order to evaluate quality of binary segmentation we selected Precision, Recall, Accuracy [168] and the Dice coefficient [231] as performance measures. These metrics are based on the combination of the number of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) classified pixels.

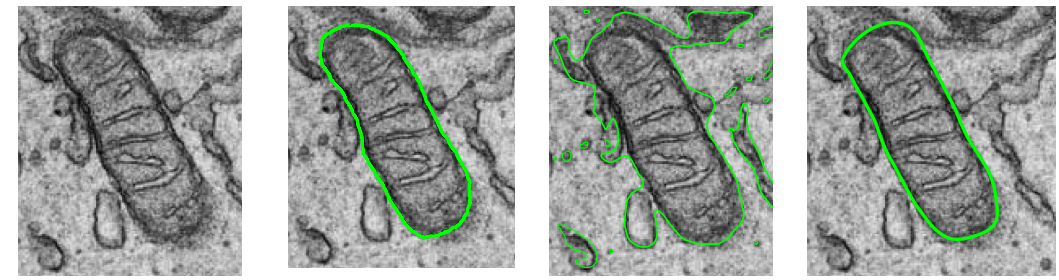
$$Precision = \frac{TP}{TP + FP} \tag{5.7}$$

$$Recall = \frac{TP}{TP + FN} \tag{5.8}$$

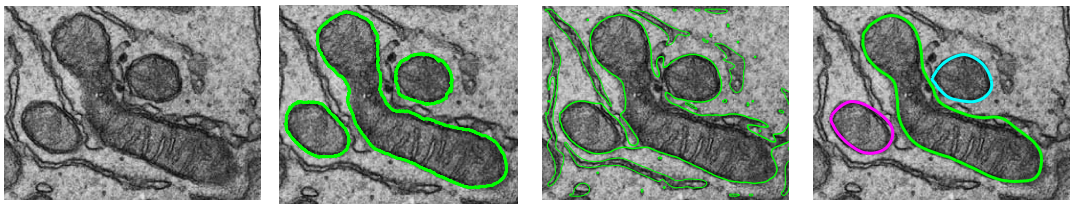
$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{5.9}$$

$$Dice = \frac{2TP}{TP + FP + TP + FN} \tag{5.10}$$

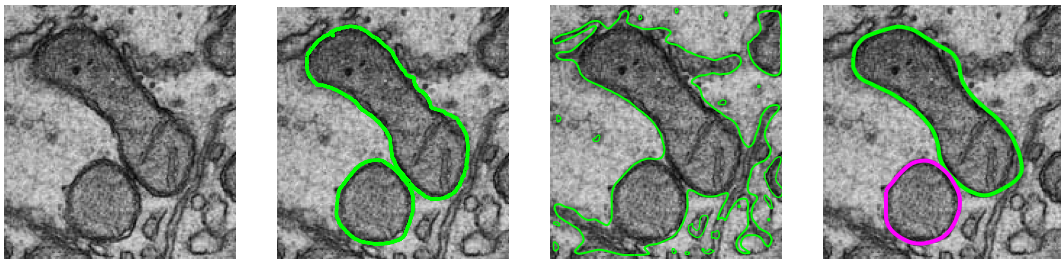
A true positive is 1 when output of the segmentation is 1 and the ground truth is 1. A true negative is 1 when the result of segmentation is 0 and the ground truth is 0. A false positive is 1 when the result of segmentation is 1 and ground truth is 0. A false negative is 1 when the result of segmentation is 0 and the ground truth data is 1. For perfect segmentation the Precision and Recall is 1. Oversegmentation is the case when Precision is low.



(a) **637**x**587** image 1 (b) Accuracy = **97**.**19**%



(c) **775**x**1031** image 9 (d) Accuracy = **95**.**81**%



(e) **758**x**710** image 17 (f) Ground Truth (g) Our after $S^{(1)}$ (h) Accuracy = **96**.**98**%

Figure 5.4.2: Selected qualitative results for the ASCB Dataset I. (a,c,e): original Electron Microscopy images; Column (f): clinically validated Ground Truth segmentation; Column (g): Our intermediate figure-ground segmentation results after the first clustering stage; **(b,d,h)**: Our final segmentation results after the contour matching step. Diagrams are best viewed in color.

Undersegmentation is the case with low Recall. Accuracy is the ratio of the sum of TN and TP to the total number of pixels. The Dice coefficient is a simple and useful measure of spatial overlap which is applied to assess the accuracy in image segmentation. The value of the Dice coefficient ranges from 0, indicating no spatial overlap between two sets of binary segmentation results, to 1, indicating complete overlap [231].

Quantitative results are shown in Fig. 5.4.5 where the Precision, Recall, Dice and Accuracy measures have been calculated with respect to the accuracy of membrane points in the segmentation and ground truth data. The quantitative results in Fig. 5.4.5 show that the average accuracy of our approach is above 90% except for the image 7 where the segmentation accuracy drops to 87.72% (Fig. 5.4.5a) and 87.19% (Fig. 5.4.5d).

The performance graph for the second dataset DII in Fig. 5.4.5(b) shows one current limitation of our approach such as the automatic selection of the *closure*-related filter size f and the downscaling factor c. The closure profile of the image 8 is shown in Fig. 5.4.5(c) and suggests

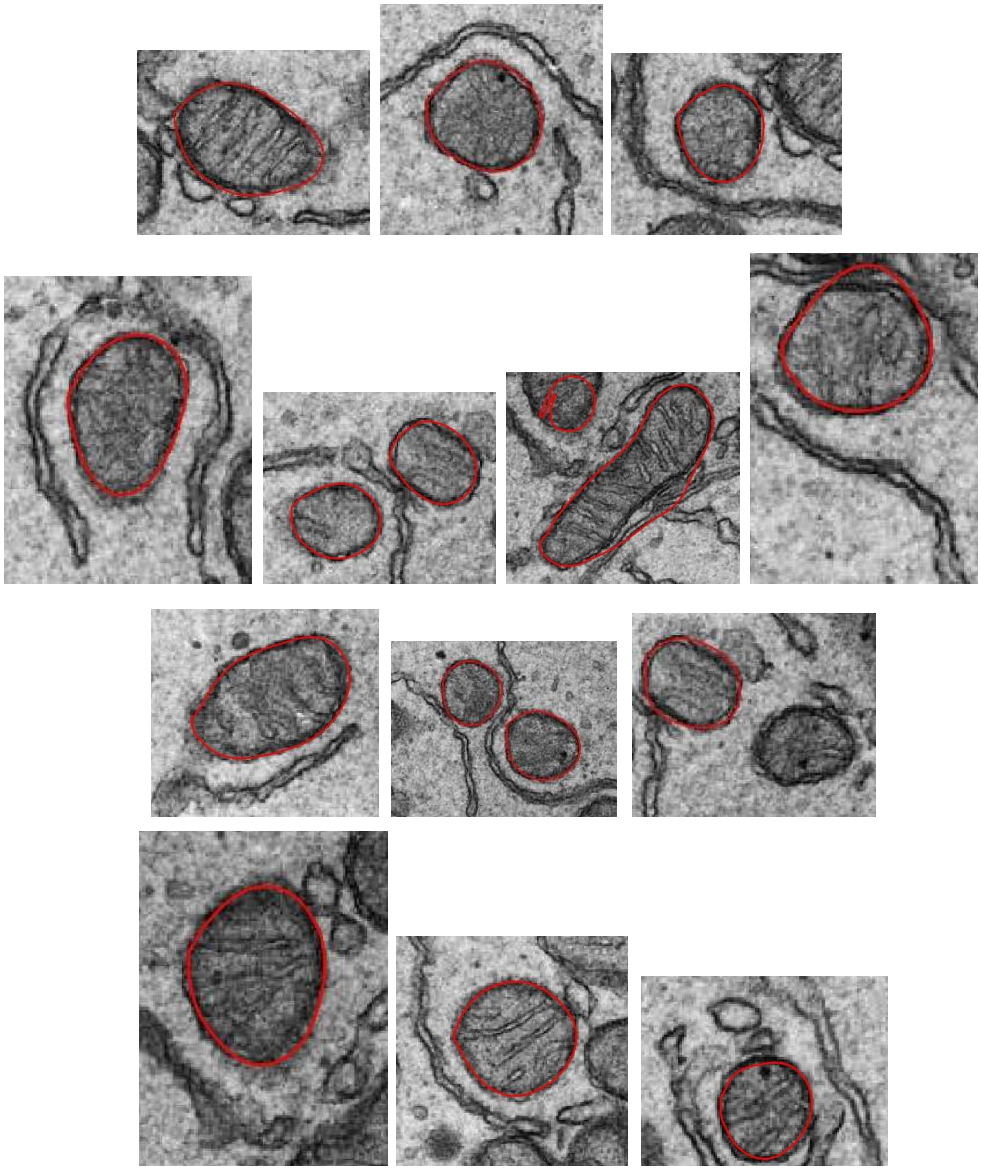
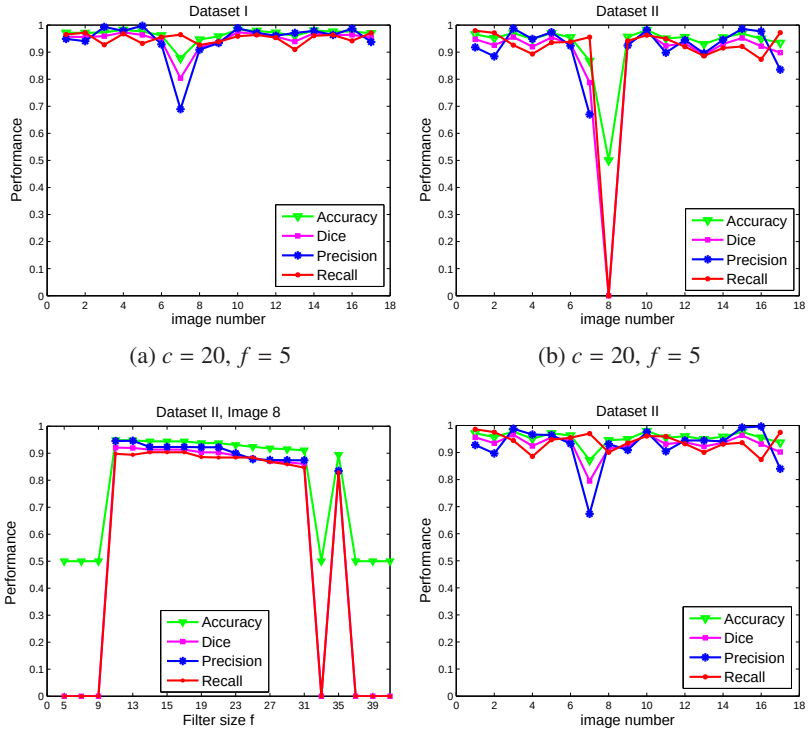


Figure 5.4.3: Segmentation results on the ASCB (American Society for Cell Biology) Dataset II where all images have different sizes but equal resolution. Quantitative segmentation results for both datasets are provided in Fig. 5.4.5. Diagrams are best viewed in color.

the appropriate filter size ranging from 9 to 31. The results with optimized c and f parameters in Fig. 5.4.5(d) show the improved performance as compared to Fig. 5.4.5(b).

This chapter presents a new approach to the unsupervised segmentation of mitochondria in EM images based on a combination of principles governing perceptual organization and unsupervised machine learning. The major theoretical novelty of our work lies primarily in the introduction of adaptive similarity models which eliminate the need for user intervention towards the selection and tuning of penalization parameters. In this way, while our segmentation algorithm runs on each image tile separately, the model parameters can be adjusted automatically. We demonstrate the successful separation of adjacent organelles with segmentation accuracy of above 90% when working with low contrast large scale electron microscopy images.

Figure 5.4.4: Selected segmentation results using the user-interactive semi-automated tracing software *Livewire* [200]. Diagrams are best viewed in color.



(a) $c = 20$, $f = 5$

(b) $c = 20$, $f = 5$

(c) Closure profile, $c = 20$

(d) $c = 15$, $f = 15$

Figure 5.4.5: Quantitative mitochondria segmentation results. **(a)** Dataset I (DI). The drop in performance for the image number 7 is due to an additional mitochondrion which has been detected by our algorithm but omitted in the ground truth segmentation. **(b)** Results for Dataset II (DII) with the parameters optimized for DI. **(c)** Closure profile for the image number 8. **(d)** Performance graph for DII with optimized c and f. Diagrams are best viewed in color.

Because we have been limited by the number of images supplied by our biomedical partners, obtaining a larger dataset along with the ground truth annotations would be beneficial for the evaluation of the method. We have approached the computational complexity of spectral clustering by subsampling and partitioning input images into a set of non-overlapping rectangles. Next chapter presents the novel Spectral Clustering Equivalence algorithm, which avoids the use of eigenvectors in the feature space, and which can be applied to very large scale, asymmetric, non-Euclidean, and generally dense datasets and similarity matrices.

Chapter 6

Anticorrelation-based Dimensionality Reduction

This chapter introduces the Spectral Clustering Equivalence (SCE) algorithm which is intended to be an alternative to spectral clustering with the objective of improving both the speed and quality of segmentation[1]. Instead of solving for the spectral decomposition of a similarity matrix as in spectral clustering, SCE converts the similarity matrix to a column-centered dissimilarity matrix and searches for a pair of the most anticorrelated columns. The orthogonal complement to these columns is then used to create an output feature vector (analogous to eigenvectors obtained via spectral clustering), which is used to partition the data into discrete clusters. We demonstrate the performance of SCE on a number of artificial and real datasets by comparing its classification and image segmentation results with those returned by kernel-PCA and Normalized Cuts algorithm. The column-wise processing allows the applicability of SCE to very large scale problems and asymmetric datasets.

This work provides a theoretical alternative to model-based spectral clustering approaches which are based on the direct eigenvector analysis – infeasible for very large and particularly dense and indefinite similarity matrices. Therefore, the method developed also directly addresses the present and inherent limitations of spectral clustering in regard to its computational complexity of $O(N^3)$. Initially the method is developed for two-class problems in view of the unsupervised foreground to background segmentation in the image space. The main research direction is set to explore second order statistics and possible alternative interpretations of similarity matrices in attempt to construct a set of orthogonal and uncorrelated components – a result achieved by the selected benchmark method of embedded kernel-PCA. A further objective is that the developed SCE algorithm be robust to the symmetry and metric violations of the pairwise proximity data and remove the unreliable dimensions more effectively than the conventional spectral clustering.

6.1 Introduction

Recent years have witnessed an enormous increase in research and applications devoted to spectral clustering where the problem of grouping is reformulated in an induced feature space. This

[1]This chapter has been published in the Lecture Notes in Computer Science series (Springer, 2011) [52]

attention is not undeserved for several reasons. Firstly, spectral clustering can be referred to as a fully *unsupervised* classification method [230]. Secondly, spectral clustering excels in discovering hidden and secondary relationships [164], managing non-convex cluster shapes, non-metric data [118] and noise reduction [101] in a well-defined and theoretically sound framework. Finally, the segmentation and grouping based on eigenvectors is able to return the perceptual organization features present in an image [147, 166, 220]. Conceptually, spectral clustering belongs to the domain of manifold learning methods aimed at the unsupervised extraction of a low-dimensional representation [201]. The term *spectral* therein refers to a broad family of clustering methods that make use of the eigenvectors of some normalized similarity matrix [7]. Different spectral clustering algorithms formalize the grouping problem in different ways, and differ widely in the number and ranking of eigenvectors and matrix normalization steps retained [155, 190]. One application to image segmentation is the spectral graph-based *normalized cuts* (Ncut) algorithm presented by Shi and Malik in 1997 as the first application of spectral clustering to computer vision domain [189].

Despite its merits, spectral clustering also has limitations associated with the computational complexity of spectral decomposition [201, 220], and the problem of discretization of continuous eigenvectors [147]. In particular, pixel classification tasks for large indefinite (possibly asymmetric) and fully dense similarity matrices form a considerable computational bottleneck for spectral clustering. Given an image with N pixels, the size of the similarity matrix increases to $N^2 \times N^2$ and the decomposition-based implementations of spectral clustering quickly become infeasible. This is a well-known fact which has been regularly emphasized over the past decade [201, 220]. Approaches dealing with this paradigm range from exploiting the sparsity, to subsampling of an image or similarity matrix, to low-rank approximation methods such as Nyström algorithm [15, 68]. We infer that this is still an open problem from the most recent work by Chen et al. [35] where the team of researchers presented a parallel HPC implementation of spectral clustering. Another work by Tung et al. [208] approached the scalability problem of spectral clustering by using a combination of blockwise processing and stochastic ensemble consensus.

In this contribution, we do not seek numeric or platform-related solutions to speed-up spectral clustering, but rather seek a method which questions the optimality of eigenvectors. We therefore raise a question if the important aspects of the data can be represented through a less expensive alternative. This consideration opens the door to a much broader range of techniques in statistical machine learning and dimensionality reduction which forms the basis for any spectral clustering implementation. Therein, the family of learning methods encompasses but is not limited to Principal Component Analysis, kernel-PCA, Linear Discriminant Analysis, other generative, discriminative, latent variable [179] and Independent Component Analysis methods [101].

In this work, we assume an underlying low-dimensional manifold and search for a computationally less expensive alternative to eigenvector-based analysis. We highlight the idea behind our algorithm that given a separable dataset [179], the core information related to the leading eigenvectors is contained in the columns of a kernel matrix. We call our algorithm Spectral Clustering Equivalence (SCE), and in the next sections we show its connection with kernel-PCA, *Ncut* and the Ising model [31]. A major contribution of our work, presented in this chapter, is the reformulation of the standard spectral clustering concept through the construction of uncorrelated, orthogonal and centered components without the use of eigenvectors.

6.2 Development of the Spectral Clustering Equivalence

In multidimensional scaling, the spectral decomposition is carried out on the inner product (Gram) matrix **G** in the feature space, with the main emphasis on preserving the inner-point distances [176]. Let us denote a similarity matrix by **S** and a dissimilarity matrix by **A**. The matrices **A** or **S** can either be viewed as a dot product matrix **G** in some feature space, according to Schölkopf and Smola [182], or transformed to **G** with kernelization and normalization [176], according to a particular spectral clustering formulation and application [220].

We assume that **A** is a generally indefinite (possibly asymmetric) matrix and interpret it as a multidimensional space spanned by its columns. Furthermore, we center the columns of **A**, call the new matrix **C** and consider a 2-class data partitioning problem. In order to answer the question about which columns in **C** carry more information about the binary class labels, we proceed with the analysis of linear dependencies present in **C**. From the related works on linear dependency analysis, Srebro and Jaakkola [194] for example, also seek to identify a low-dimensional subspace that captures the dependent and the "important" aspects of the data, and separate them from independent variations. Thus, a natural way to conduct dependency analysis is to analyze correlations between different variables, and the first step, prior to applying correlation analysis, is the centering of variables. Contrary to the formation of **G** in kernel-PCA, which involves double-centering [176], our normalization of **A** in order to obtain **C** does not involve row centering. Next, we define the correlation between two centered columns, $\mathbf{c}_1$ and $\mathbf{c}_2$, according to the formula of Pearson's product-moment correlation coefficient [170]:

$$\rho_{1,2} = \sum_{i=1}^{N} [\mathbf{c}_{1,i}\mathbf{c}_{2,i}] / \sqrt{\sum_{i=1}^{N} \mathbf{c}_{1,i}^2 \sum_{i=1}^{N} \mathbf{c}_{2,i}^2} \ . \tag{6.1}$$

A strong negative correlation provides a suitable measure of discrimination according to [207], for example, and also indicates that the decrease in one variable is controlled by the increase in the second variable. With regard to natural images, it is reasonable to view foreground and background as the two most distinct and thus most anticorrelated image structures. We therefore are interested in the lower bound of $\rho \in [-1, 1]$, and define a pair of observations (columns of **C**) with a strong negative correlation $\rho_{ij} \to -1$ dissociation patterns.

In order to construct the orthogonal and uncorrelated kernel-PCA estimates we first draw on the idea of canonical spaces [197]. In statistics, the canonical (or *principal*) angles are closely related to the measure of dependency and covariance of random variables. When applied to column spaces of matrices, the principal angles describe canonical correlations of a matrix pair [107].

Let $\mathcal{X}$ and $\mathcal{Y}$ be the two unknown subspaces spanned by the columns of **C**. The largest canonical angle $\theta(\mathcal{X}, \mathcal{Y})$ between $\mathcal{X}$ and $\mathcal{Y}$ is defined according to [197] as:

$$\theta(\mathcal{X}, \mathcal{Y}) = \max_{x \in \mathcal{X}} \min_{y \in \mathcal{Y}} \angle(x, y) \tag{6.2}$$

We know that the correlation coefficient between centered variables is equivalent to the cosine of the angle between these variables [170]. The cosine of the largest canonical angle $\theta \to \pi$ can therefore be interpreted as the minimal signed Pearson correlation coefficient $\rho_{min} \to -1$ between a pair of mostly anticorrelated columns of **C**.

input					output
image →	features →	$\mathbf{S} \rightarrow \mathbf{A}$ →	Dimensionality Reduction →	$X_{1,2}$ (**SC**) $\Phi_{1,2}$ (**SCE**)	set of discrete clusters

$$\mathbf{A} \rightarrow \mathbf{G} = \mathbf{U}\Lambda\mathbf{U}^T \rightarrow \lambda_{min} \rightarrow \tilde{\mathbf{G}} = \tilde{\mathbf{U}}\tilde{\Lambda}\tilde{\mathbf{U}}^T \rightarrow X = \tilde{\mathbf{U}}\sqrt{\tilde{\Lambda}} \rightarrow X_{1,2} \ . \tag{6.3}$$

$$\mathbf{A} \rightarrow \mathbf{C} \rightarrow \mathbf{R} \rightarrow \rho_{min} \rightarrow \mathbf{c}_{1,2} \rightarrow^{\perp} \rightarrow \mathbf{t}_{1,2} \rightarrow \mathbf{M}\mathbf{t}_{1,2} \rightarrow \begin{matrix}\text{decorrelate}\\ \text{with } 2\times 2 \text{ PCA}\\ \text{and center}\end{matrix} \rightarrow \Phi_{1,2} \ . \tag{6.4}$$

Figure 6.2.1: Algorithm description and the comparison of spectral clustering (6.3) and SCE (6.4). Both methods take a similarity matrix $\mathbf{S}$ as an input and produce a pair of orthogonal and uncorrelated feature vectors. Spectral clustering is based on the eigendecomposition of the pseudo-Gram matrix $\mathbf{G}$. Conversely, SCE is a decomposition-free method which is based on the dependency analysis of the column-centered kernel matrix $\mathbf{C}$. Because SCE works on the columns of $\mathbf{C}$, it can generally be applied on asymmetric, non-PSD and rectangular datasets.

In the next step, we construct the symmetric matrix $\mathbf{R}$ with entries corresponding to the correlation coefficients ρ_{ij} computed between the columns i and j of the matrix $\mathbf{C}$. According to the cosine definition, π is the maximum possible angle corresponding to $\cos(\theta) = \rho_{min} = -1$ [170]. Thus, ρ_{min} not only defines a pair of mostly anticorrelated columns $\mathbf{c}_1$ and $\mathbf{c}_2$, but also provides the link with the first canonical angle θ_1.

According to Stewart [197], the number of canonical angles in the case of $\dim(\mathcal{X}) < \dim(\mathcal{Y})$ is equal to $\dim(\mathcal{X})$, which in our case is $\dim(\mathcal{X}) = 2$. This fact allows the construction of the second SCE-based feature component by employing the orthogonality constraint to obtain $\theta_2 \rightarrow \pi/2$. For this purpose and $\forall k \in [1, N^2]$ we seek an orthogonal complement vector $\mathbf{c}_k$ to $\mathbf{c}_{1,2}$, and in this process we discard the least orthogonal pair of vectors:

$$\mathbf{t}_{1,2} = \begin{cases} \mathbf{c}_{1,k}, & \text{if } |\angle\mathbf{c}_1, \mathbf{c}_k - \pi/2| < |\angle\mathbf{c}_2, \mathbf{c}_k - \pi/2| \\ \mathbf{c}_{2,k}, & \text{otherwise} \ . \end{cases} \tag{6.5}$$

This gives us a pair $\mathbf{t}_{1,2}$ of somewhat orthogonal columns of matrix $\mathbf{C}$ where $\rho(\mathbf{t}_1, \mathbf{t}_2) \rightarrow 0$. It is a rather strong assumption that the $\mathbf{t}_{1,2}$ obtained would result in $\rho = 0$. Furthermore, we multiply $\mathbf{t}_{1,2}$ by $\mathbf{C}^z$ (i.e. $\mathbf{C}$ to the power of z) to maximize linear dependency and subsequently decorrelate and center using PCA on the computed 2×2 covariance matrix. By doing so, we obtain the pair $\Phi_{1,2}$ of completely orthogonal, uncorrelated and centered SCE feature vectors.

In this formulation, the SCE approximation of kernel-PCA feature vectors $\mathbf{X}_{1,2}$ is controlled by the coefficient z. After decorrelation and centering, and similarly to the approach used in the ranking of PCA components [101], our selection of the leading feature vector Φ_1 is based on the maximum variance principle such as $\sigma^2(\Phi_1) > \sigma^2(\Phi_2)$ where σ denotes standard deviation.

The complete description of the SCE algorithm, its comparison with the selected kernel-PCA based spectral clustering implementation and the progression from $\mathbf{c}$ to Φ is outlined in Fig. 6.2.1.

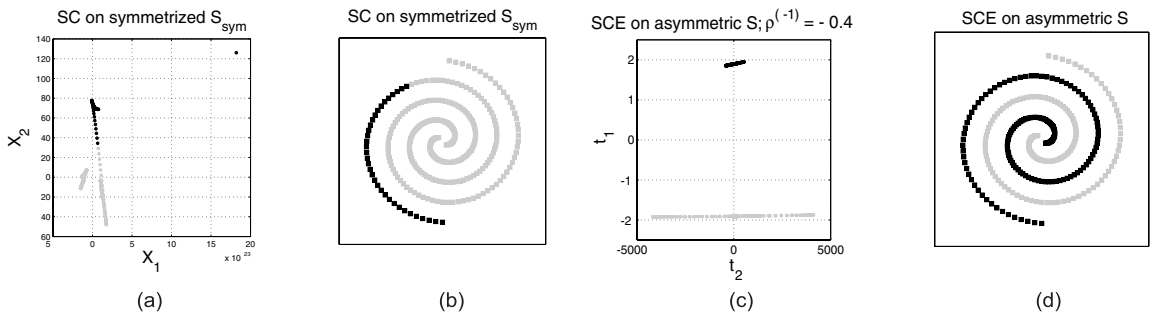


Figure 6.3.1: Comparison of classification results on asymmetric data with binary k-means in spectral clustering (a) and SCE (c) constructed feature spaces. An interlocked two spirals dataset is considered as a challenging benchmark for spectral clustering [33]. As can be seen in (c), SCE results in a better inter-cluster separation in the feature space. As the symmetry condition constitutes one of the four metric axioms [9], this example tests the non-metric invariance of SCE.

6.3 Results of Comparison to kernel-PCA

In this section we compare the SCE-based classification to the kernel-PCA based result and applied them to an experiment with an asymmetric dataset. We used an interlocked spirals dataset, shown in Fig. 6.3.1, which is considered to be a challenging benchmark for spectral clustering [33]. To tackle this challenging clustering problem, Chang and Yeung [33], for example, presented a robust path-based spectral clustering algorithm with the use of a Gaussian kernel. Nearest neighbor-based grouping methods such as, for example, single linkage hierarchical clustering with the simple Euclidean distance metric are also known to solve this type of a problem. Throughout this chapter we use the k-means algorithm for clustering purposes in both spectral clustering and SCE returned feature spaces. The main objective of our experiment was to test the asymmetric and non-metric invariance of SCE and thus we designed the following asymmetric similarity measure:

$$\begin{aligned} s_{ij} &= b_1 \cdot (dx_i - dx_j), \\ s_{ji} &= b_2 \cdot (dy_i - dy_j) \ . \end{aligned} \tag{6.6}$$

In the experiment outlined in Fig. 6.3.1, each spiral consists of 151 data points and is generated according to the equation of Archimedean spiral. Two separately computed coordinate vectors have been further concatenated to form x and y vectors. In (6.6) we use the differentials dx and dy of the raw coordinates x and y. The model parameters b_1 and b_2 control the degree of asymmetry and with $b_1 = 20$ and $b_2 = 2$ we obtain a highly asymmetric $\mathbf{S}$ with $s_{ij} \neq s_{ji}$. Due to the symmetric formulation of spectral clustering we decompose $\mathbf{S}$ into its symmetric and skew-symmetric parts $\mathbf{S} = \mathbf{S}_{sym} + \mathbf{S}_{skew}$ according to [38]. Because of its symmetric formulation, the kernel-PCA based implementation disregards $\mathbf{S}_{skew}$ and diagonalizes only the remaining symmetric matrix $\mathbf{S}_{sym}$.

Given a high degree of induced asymmetry ($b_1 \gg b_2$), the combination of spectral clustering with k-means failed to correctly identify the two separate spirals as illustrated in Fig. 6.3.1(b). Conversely, SCE fully utilized the information in the asymmetric component of $\mathbf{S}$ to achieve the correct separation as shown in Fig. 6.3.1(d). Furthermore, SCE with k-means on the asymmetric $\mathbf{S}$ resulted in a better projection and thus higher inter-cluster separability than kernel-PCA. In this experiment, the anticorrelation amounted to the global value of $\rho_{min} = -0.4$ in the matrix $\mathbf{R}$.

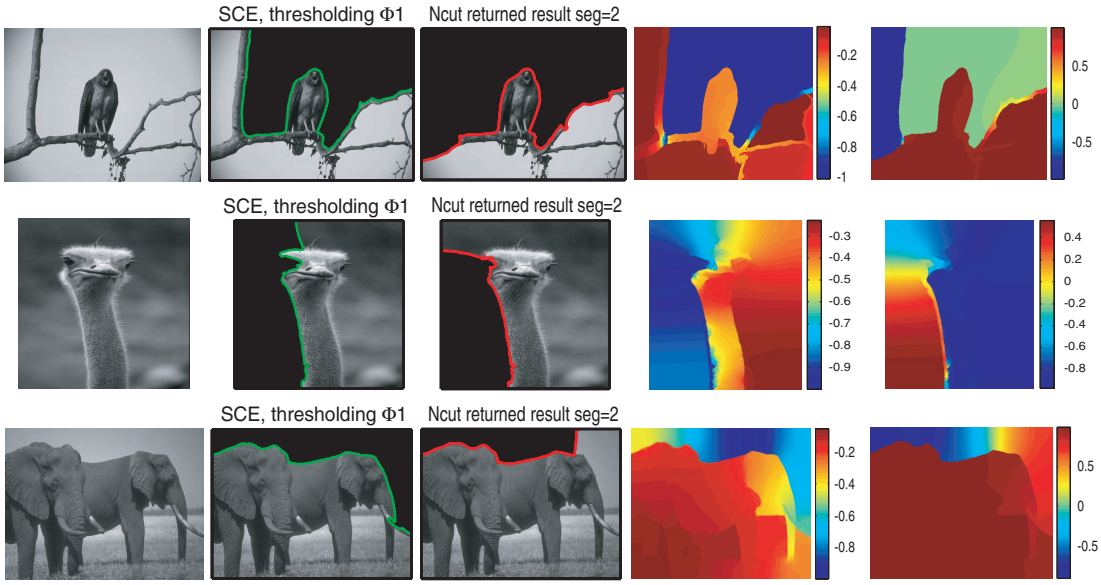


Figure 6.4.1: Original 321×481 images from the Berkeley database are shown in the first column and SCE-based results in the second column. The last three columns show the *Ncut*-based result and the first and the second eigenvector returned by *Ncut*. This diagram is best viewed in color.

6.4 Connection of SCE with Normalized Cuts

Ncut is the graph-theoretic formulation of spectral clustering with the objective of minimizing a normalized measure of disassociation [190]. *Ncut* operates on the 2nd generalized eigenvector of a normalized weight matrix $\mathbf{W}$ where the normalization procedure has the purpose of penalizing large image segments. *Ncut* then computes the diagonal matrix $\mathbf{D}$ containing the sum of all edges, and solves for the eigenvectors of $\mathbf{N} = \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$ with $\mathbf{N}(i,j) = \mathbf{W}(i,j)/\sqrt{\mathbf{D}(i,i)}\sqrt{\mathbf{D}(j,j)}$. The second smallest generalized eigenvector λ_2 of $\mathbf{W}$ is a componentwise ratio of the second and first largest eigenvectors of $\mathbf{N}$ [220]. We are interested in whether or not our Φ-based approximation can provide computational savings over the *Ncut* while maintaining the same binary image partition.

For comparative purposes, we have acquired the *Ncut* demo software from [40] and used the parameters supplied for the calculation of the adjacency matrix based on intervening contour similarities. In order to compare the segmentation results, we take the $\mathbf{N}$ matrix returned and compute Φ as outlined in Fig. 6.2.1. There are two aspects which are non-trivial in connection with *Ncut*: definition of a feature similarity, and selection of a partitioning threshold which can take the values of 0, median or a point that minimizes $Ncut(A,B)$ [190]. In our SCE formulation we center the columns of $\mathbf{N}$ before mixing with $\mathbf{t}$ for a number of z iterations. We can also implement $\mathbf{N}^z\mathbf{t}$ mixing iteratively by centering only the mixed $N^2 \times 2$ components after each iteration. The successive centering results in the ideal non-parametric case, where we partition the graph according only to the signs $(A = \{\Phi_1 > 0\}, B = \{\Phi_1 \leq 0\})$.

For our experiments we used a Dell Precision M6300 dual-core notebook with 2GB RAM and the Matlab R2009a environment. The results of an experiment on the full resolution 321×481 images from the Berkeley segmentation database are shown in Fig. 6.4.1. The binary *Ncut* partition returned (Fig. 6.4.1, third column) is given by the second computed eigenvector (fifth column).

We observe the qualitative equivalence between the SCE segmentation based on Φ_1, and the first eigenvector (fourth column) returned by the *Ncut* algorithm, which also outputs a very narrow-banded sparse matrix $\mathbf{N}$ with $\approx 0.1\%$ non-zero elements. Due to the inherent sparsity advantage which has its roots in the definition of similarities [190], *Ncut* does not rely on the direct eigen-decomposition of $\mathbf{N}$. Instead, it uses the iterative Lanczos eigensolver [197] which, similarly to SCE, is also based on sparse matrix-vector multiplications. In our experiments with the returned sparse matrix $\mathbf{N}$, the smallest correlation coefficient ρ_{min} is marginally low ($\rho_{min} = -8.6025e-004$) which explains the high number of SCE iterations ($z \approx 1 \times 10^4$) needed to approximate the first *Ncut* eigenvector.

6.5 SCE Extension with Latent Variables

It is known that eigenvectors are efficient in capturing perceptual organization features [147, 166, 220]. *Ncut* is also based on the view that perceptual grouping should be a process that aims to extract global impressions of a scene, and it provides a hierarchical description of it [190]. In our SCE formulation, the approximation to the first *Ncut* eigenvector is connected with the rotation and scaling of two hyperplanes, implemented through iterative $\mathbf{N} \times \mathbf{t}$ multiplications. Therefore, it is reasonable to assume that the foreground innovations can be extracted ahead of the eigenvector equivalence condition and in a much shorter time. To approach these problems, we followed the idea of a 'greedy search' and designed the optimization procedure detailed in Table 6.5.1.

Table 6.5.1: Ising-based SCE extension.

function $[\Omega, A, B]$ = scecut($\mathbf{N}$, $\mathbf{t}$, itmax)
$\mathbf{k}$ = cov($\mathbf{t}$); $[\mathbf{u}, \mathbf{v}]$ = eigs($\mathbf{k}$); $\mathbf{t} = \mathbf{t} * \mathbf{u}$; $\mathbf{t} = \mathbf{t} - \bar{\mathbf{t}}$; it = 0;
while it<itmax
  $\mathbf{b} = \mathbf{N} * \mathbf{t}$; $\mathbf{b} = \mathbf{b} - \bar{\mathbf{b}}$;
  if std($\mathbf{b}_1$) > std($\mathbf{b}_2$)
    $\mathbf{t}_1 = -\text{sign}(\mathbf{b}_1) * \mathbf{b}_2$; $\mathbf{t}_2 = \mathbf{b}_2$;
  else
    $\mathbf{t}_1 = -\text{sign}(\mathbf{b}_2) * \mathbf{b}_1$; $\mathbf{t}_2 = \mathbf{b}_1$;
  end
  it = it + 1;
end
$\Omega(:, 1) = \mathbf{t}_1$; $\Omega(:, 2) = \mathbf{t}_2$

$A : \Omega(:, 1) > 0$; $B : \Omega(:, 1) \leq 0$;

Here, we view the matrix $\mathbf{N}$ as the matrix of features and consider $\mathbf{N}$ to be sparse. We view the signs of a pair of columns as two latent (hidden) binary support variables, and thus establish the connection with the Ising model which is a special Markov Random Field [31].

In Table 6.5.1, "std" denotes the standard deviation. Instead of using the Ising model to represent pixels, we work with similarities contained in the normalized sparse matrix $\mathbf{N}$. We denote the optimized feature vectors by Ω.

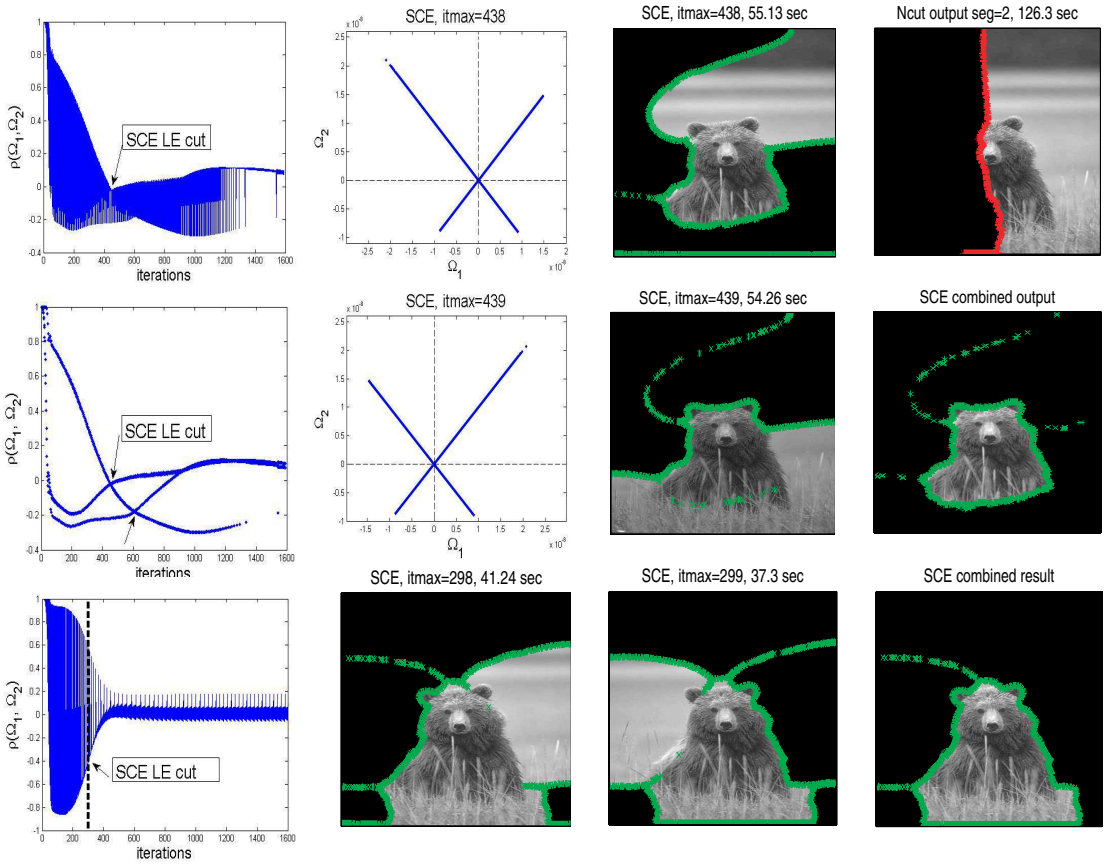


Figure 6.5.1: Concept of the Ising-based SCE. We analyze the dynamic oscillatory behavior of the correlation coefficient to find the optimal transition. For the two iterations near the transition point we compute the binary classification and combine the results to yield the optimal figure ground segmentation. The transition is given by the point where the hyperplanes are flipped around the Ω_1 (strongest) axis as can be seen in the second column. On our computer, SCE runs faster than *Ncut* (see the computational time above the diagrams), and returns more perceptually meaningful binary segmentation. Diagrams are best viewed in colour.

The initial condition is given by the pair of columns $\mathbf{t}_{1,2}$ which, with the change in notation such that $\mathbf{S} \equiv \mathbf{N}$, can be obtained according to:

$$\mathbf{N} \rightarrow \mathbf{P} \in \mathbb{R}^{N^2 \times H} \rightarrow \mathbf{p}_i = \mathbf{p}_i - \bar{\mathbf{p}}_i \rightarrow \mathbf{R} \rightarrow \mathbf{c}_{1,2} \rightarrow \mathbf{t}_{1,2} \quad . \tag{6.7}$$

For the experiments in this Section we used Matlab R2009a with 2GB RAM to process images from the Berkeley database. Because the image size is 321×481 pixels, it was not possible to store the resulting similarity matrix in Matlab. Due to these limitations we did not search for the global minimum on correlation in $\mathbf{N}$, but instead operated on a subset matrix $\mathbf{P}$ of H randomly computed columns of $\mathbf{S}$. Thus, in the experiment shown in Fig. 6.5.1 we selected the reduced subspace of $H = 100$ in order to process a 321×481 image from the Berkeley database. Although the automatic selection of the stopping criterion is still an ongoing work, we note that one possibility to obtain the optimal partition is to analyze the dynamic oscillatory behavior of the correlation coefficient $\rho(\mathbf{t}_1, \mathbf{t}_2)$ (first column in Fig. 6.5.1) and we observed that the optimal figure-ground cut occurs at the change in phase of ρ. The binary A (figure) and B (ground) partitions have to be computed twice for the two successive iterations corresponding to the ρ transition.

The final segmentation result F combines the intermediate results at different iterations such that $F = F_1 \cap F_2$, where $F_1 = A_1 \cup B_1$ and $F_2 = A_2 \cup B_2$ (see Fig. 6.5.1 last row).

Random subsampling of $\mathbf{N}$ (6.7) explains somewhat different (but consistent with perceptual meaning) segmentation results in the second and the third row of Fig. 6.5.1, where we used different random subsets of $\mathbf{N}$. The results in Fig. 6.5.1 show that the Ising-based SCE algorithm developed in this work not only detects the foreground innovations in the analyzed image, but also has a factor of two speed-up compared to *Ncut* algorithm.

To summarize, in this chapter we have developed an alternative to eigenvector-based feature classification. We started the research by examining the conditions of the feature space equivalence between the proposed SCE and the kernel-PCA outputs. We have shown that the proposed algorithm reduces the dimension of the feature space while improving classification performance and thus results in a better projection and higher inter-cluster separability than kernel-PCA.

With regard to image segmentation, we demonstrate that the proposed method has potential to replace eigenvector-based computation at least for applications considering the detection of foreground innovations. Our future work will concentrate on generalizing SCE to multiclass problems as well as investigating the regularization and stopping criteria of the proposed Ising-based SCE extension. Although the Ising model takes SCE beyond the equivalence pursuit, it shows that segmentation without eigenvectors is a more flexible framework than that offered by the standard spectral clustering.

Chapter 7

Thesis Conclusion and Future Work

The aim of this chapter is to highlight the main contributions associated with this research and to provide conclusions in relation to the objectives of this thesis. In the second section, future work and its significance for this research is outlined and examined.

7.1 Contributions and Conclusions

In this work, we attempt to unify different lines of research drawn from the areas of perceptual grouping, machine learning, image processing, kernel-based dimensionality reduction and spectral clustering. This thesis aimed to primarily deliver the solution to the automatic localization and segmentation of mitochondria in subcellular electron microscopy images and was cast within the framework of machine learning. Specifically, we adopted the spectral clustering approach which belongs to the family of unsupervised machine learning tools. Three major contributions [51, 50, 52] have emerged from this research and are summarized below.

Firstly, we considered the problem of modeling and extracting lamellar and tubular mitochondrial morphologies defined by the inner membrane folding. The aim was to obtain a localization marker which could be used to extract the outer contour of a mitochondrion. This was the point where we first diverted our attention to the grouping of low-level primitives by using Gestalt laws of visual segregation or perceptual grouping. We have observed that in a network of line segments extracted from microscopic images, the mitochondrial structure of the above morphologies can be captured through a new grouping principle of *Projectivity*. Furthermore, we have investigated properties of the proposed similarity measure and have shown how to cluster line segments with the embedded kernel-PCA and also the single linkage hierarchical clustering approaches. The discussion on the experimental findings for this topic has been provided in Chapter 4.

Next, we extended the idea of grouping line segments to the grouping of pixels in gray scale microscopic images with the goal of segmenting mitochondrial membranes. We have observed that we can use such perceptual grouping cues as *Similarity*, *Proximity* and *Figure-Ground* for the construction of the similarity measure. The major contribution is therefore an unsupervised segmentation framework (proposed in Chapter 5) which follows the evolutional ability of human vision to extrapolate significant structures in an image. In Chapter 5 we adopt a perceptual grouping strategy by selecting the spectral clustering framework, which is known to capture perceptual

organization features, as well as by developing similarity models according to the Gestalt laws of visual segregation. Our proposed framework can be applied to the detection of cells and organelles in microscopic images. The recursive spectral clustering processing pipeline is the main building block of our method and consists of preprocessing, two-stage binary image segmentation and subsequent contour matching.

Another theoretical contribution of this work resides in the formulation of robust similarity models which automatically adapt to the statistical structure of the biological domain, and owes optimal performance in pixel classification tasks under the wide variety of distributional assumptions. This has been achieved by introducing proper normalization of input variables and subsequent data fusion to yield adaptive similarity measures. Such adaptive similarity models eliminate the need for the manual tuning of model parameters. Full size high-resolution images can not be directly processed due to the resulting size of the similarity matrices and therefore we had to downsample and tile the input image in the preprocessing step. Because of the adaptive nature of the similarity models introduced, there is no need to adjust the model parameters in each tile. The experimental results have shown segmentation accuracy above 90% on average, but more data and ground truth is desirable for evaluation. The expensive nature of spectral decomposition has motivated us to develop a new eigenvector-free Spectral Clustering Equivalence algorithm that can output orthogonal and uncorrelated components equivalent to that of the selected spectral clustering algorithm based on the embedded kernel-PCA.

We have addressed the computational limits of spectral clustering in the third major contribution of this thesis presented in Chapter 6. Here, we first followed the intuition that in the model-based spectral clustering formulation, physically meaningful signals are already contained in the kernel or Gram matrices, and the correct cluster assignments can be inferred by higher order statistical analysis. This work provides an alternative to model-based spectral clustering approaches which are impractical for large, particularly dense, asymmetric and generally indefinite similarity matrices. The major contribution here resides in the introduction of the SCE (Spectral Clustering Equivalence) algorithm which is intended to be an alternative to spectral clustering with the objective of improving both speed and quality of segmentation. Instead of solving for the spectral decomposition of a similarity matrix as in spectral clustering, SCE converts the similarity matrix to a column-centered dissimilarity matrix and searches for a pair of the most anticorrelated columns. The orthogonal complement to these columns is then used to create an output feature vector (analogous to eigenvectors obtained via embedded kernel-PCA), which is used to partition the data into discrete clusters. We demonstrated the performance of SCE on a number of artificial and real datasets by comparing its classification and image segmentation results with those returned by the embedded kernel-PCA and graph-based Normalized Cuts algorithm. The column-wise processing allows the applicability of SCE to very large scale problems and asymmetric datasets.

In this contribution we did not seek numeric or platform-related solutions to speed-up spectral clustering but rather a method which questions the optimality of eigenvectors. We therefore raised the question of whether the important aspects of the data can be represented through a less expensive alternative. This consideration opens the door to a much broader range of techniques in statistical machine learning and dimensionality reduction which form the basis for any spectral clustering implementation. Therein, the family of learning methods encompasses, but is not limited

to PCA, kernel-PCA, Linear Discriminant Analysis, other generative, discriminative, latent variable and Independent Component Analysis methods. We assumed an underlying low-dimensional manifold and searched for a computationally less expensive alternative to eigenvector-based analysis. We next highlighted the idea behind our algorithm that given a separable dataset, the core information related to the leading eigenvectors is contained in the columns of a kernel matrix. We also show the connection of the proposed SCE with embedded kernel-PCA, *Ncut* and the Ising model. The major contribution presented in Chapter 6 is thus the reformulation of the standard spectral clustering through the construction of uncorrelated, orthogonal and centered components without the use of eigenvectors.

In summary, the work presented in Chapter 6 provides a theoretical alternative to model-based spectral clustering approaches which are based on the direct eigenvector analysis – infeasible for very large and particularly dense and indefinite similarity matrices. Therefore, the developed method also directly addresses the present and inherent limitations of spectral clustering in regard to its computational complexity. Initially the method has been developed for two-class problems in view of the unsupervised foreground to background segmentation in the image space. Based on inferences and improvements made therein, the problem is expected to be generalized to multiple classes. The main research direction explored second order statistics and possible alternative interpretations of similarity matrices in an attempt to construct a set of orthogonal and uncorrelated components – a result achieved by the selected benchmark method of embedded kernel-PCA. The method developed proved to be robust to non-metric violations and was able to remove the unreliable dimensions more effectively than the conventional spectral clustering.

At the very beginning of this research, our choice of using supervised machine learning was constrained by the amount of data provided by our biomedical partners. All our developed algorithms would benefit from larger datasets, both raw and annotated ground truth images. It can be seen in a number of publications that shape- and texture-based classifier training on the annotated images is able to extract mitochondria of arbitrary morphologies in pre-defined tissue and cell type. In a subcellular setting with well-separated mitochondria, such methods have been shown to succeed in the detection and delineation of organelles. However, we note that in case of apoptotic tissues, these approaches may not work because the mitochondrial morphologies change from image to image with a time. To conclude, the mitochondria segmentation quest and the selection of the spectral clustering framework have proven to be a very interesting, and challenging, but also inspiring combination as it resides at the interface of computer vision, theoretical machine learning, biology and perceptual organization properties of the evolutional human vision.

7.2 Future Work

While the objectives of this study have been accomplished, some future work can be considered to extend the ideas presented in this research thesis. In this section, we discuss possible future lines of research regarding localization and segmentation of mitochondria, and the anticorrelation-based dimensionality reduction approach developed in this thesis. We also address the open problems of automatic selection of the number of clusters in the clustering algorithms used.

7.2.1 Clustering Projected Line Segments

In relation to the first contribution, we have shown that our line grouping method together with the *Projectivity* cue introduced can be used to identify hulls of mitochondrial structures in subcellular environments. However, in future the validation of our method could benefit from a larger database of mitochondria images and their ground truth segmentation for quantitative comparisons with other line grouping methods. We have noted that the resulting non-linearly separable feature points arrangements are due to the fact that the selected kernel-PCA clustering framework may be ill-posed for this problem, and that "*tracing*" should be used instead. We applied "*tracing*" of line segments by choosing single linkage hierarchical clustering on the dissimilarity matrix **A**. The future work here could focus, for example, on the development of other different kernel functions in order to obtain more linearly separable clusters in the feature space in the kernel-PCA context.

We have addressed the applicability of our proposed similarity measure to the grouping of co-linear and co-circular line segments in Section 4.9. The analysis of existing algorithms for contour integration problems and the development of a more general line grouping method could also be an interesting direction for future work to pursue. Our experimental results on clustering of line segments for the purpose of mitochondria localization showed that Single Linkage Hierarchical Clustering outperformed kernel-PCA. However, the automatic selection of the k numbers of clusters in the k-means algorithm as well as the automatic selection of the dendrogram threshold for Single Linkage Hierarchical Clustering could also be addressed in future work. Roth et al. [176], for example, also reported the problem of selecting the correct number of clusters. To alleviate this problem, the authors used the concept of cluster stability, which has been introduced by Dudoit and Fridlyand [59] and further refined by Lang et al. [115].

In our experiments we have assumed that a mitochondrion is always present and that the internal structure represented by line segments, is consistent. Everitt et al. [64] highlighted the fact that the logical starting point for a cluster analysis would be a test for the absence of cluster structure. Such tests are not usually employed in practical applications of clustering. However, if they are aimed at detecting an unknown underlying structure, then testing becomes more relevant. What is required is a model that describes the data-generating process in the absence of clustering, and a test statistic which will reflect departures from the model.

In relation to Single Linkage Hierarchical Clustering, a possible future line of research could, for example, investigate the dynamic tree cutting introduced by Langfelder et al. [116]. This method allows for different branches of the tree to be cut at different levels. The process iterates until the number of clusters is stable by combining and decomposing clusters, making successive cuts of the sub-dendrograms within clusters based on their shape. For the nested structures inherent in hierarchical trees, there are two relevant works by Duda and Hart [58] and Beale [13]. Both methods are based on the ratio of between-cluster to within-cluster sums of squares, when the cluster optimality is divided into two [64].

Radon-like features, which also involve computation of line segments, were proposed in the literature to leverage both the texture and the geometric information present in microscopy images to segment mitochondria. Future experiments could involve the computation of such descriptors and their performance comparison to our line-grouping method.

7.2.2 Mitochondria Segmentation

While working on the spectral clustering-based segmentation framework for mitochondria, we have been limited by the number of images supplied by our biomedical partners. Thus obtaining a large dataset along with the ground truth annotations would be beneficial for the evaluation of the method in the future. This would also provide us with the opportunity to test supervised machine learning tools such as SVM and CNN. The selection of the optimal k numbers of clusters was not the issue in this contribution because we considered the binary segregation of an image into foreground and background. We have approached the computational complexity of spectral clustering mainly by sub-sampling and further partitioning of input images into a set of non-overlapping rectangles. For example, Ghita et al. [76] developed a spectral clustering framework based on the Affinity Factorization algorithm [166]. In particular, the authors explored the concept of clustering superpixels instead of pixels in an image. This approach substantially reduced the complexity of the input data. We also recommend using the feature enhancement techniques in the preprocessing step in order to improve the quality of segmentation.

Other line of research could focus on an improved strategy for the integration of perceptual grouping cues, such as *continuation*, in order to obtain more accurate membrane segmentation in noisy and low contrast microscopic images. One example of related work in the context of spectral clustering, is given by Kaynig et al. [103] on the segmentation of neuronal structures in ssTEM images. Because of varying mitochondrial morphologies, we have initially targeted the global perceptual appearance properties of membranes in electron microscopy images. Our affinity models are pixel-based and therefore are limited in describing directional features. Therefore, in order to enforce the gap completion of disconnected membranes, our affinity model could benefit from incorporating, for example, the flux of the gradient vector field into the segmentation. Many other directional feature extraction methods and affinity models exist, and testing them on our datasets could also be within the scope of future research. This topic is worth of extension and further investigation.

7.2.3 Anticorrelation-based Dimensionality Reduction

The third contribution addressed the computational limitations of spectral clustering. We have initially assumed the underlying low-dimensional manifold and thus attempted to construct only the first and the second SCE feature vector. Therefore, we could investigate synthetic and real-life scenarios where the cluster information is coded by more than two eigenvectors.

Also, there is an unanswered question about why the information about the correct cluster assignments and the equivalence to higher ranked eigenvectors is contained in the most anticorrelated columns of the normalized similarity matrix. It appears that maximizing anticorrelation would be a direction to obtain more informative feature vectors. But, in the case of perfect anticorrelation, knowing $\mathbf{a}$ gives us immediately $\mathbf{b} = -\mathbf{a}$, where $\mathbf{a}$ and $\mathbf{b}$ are perfectly anticorrelated. There is no gain in information apart form the sign. What is desirable here, is a logical explanation and quantitative analysis of the "*information*" term. Two information maximization-based approaches have been proposed by Bell and Sejnowski [16] and by Ralph Linsker [126]. We have discussed these approaches in Section 2.2.8 in the context of unsupervised neural networks. Both works could be

relevant for the explanation of our anticorrelation-based dimensionality reduction concept.

The proposed SCE framework would benefit from the further investigation of the computational algorithm behavior on dense as well as sparse similarity matrices. We highlight that in cases where very large scale matrices can not even be fully computed and stored, the SCE algorithm has the advantage of working on a small subspace of a dense similarity matrix. However, we still do not know if this strategy can be applied on a sparse matrix, and thus more insight is needed in the future work. More experiments could be designed in the future to address the classification of synthetic datasets with additive noise, clustering of line segments extracted from real images containing mitochondria, and quantitative and computing time comparisons between SCE and standard kernel-PCA techniques.

Through our background research, we have seen that there are a number of different dimensionality reduction and clustering algorithms which are based on neural networks. An interesting line of future research could investigate their suitability to the localization and segmentation of mitochondria and comparison with our proposed methods. Experiments could include comparison between self-organizing maps and k-means, for example, in classification of non-linearly separable datasets. A relevant work on the kernel topographic map formation is given by Bishop et al. [21, 22] where the authors presented the Generative Topographic Mapping algorithm. Van Hulle [213] applied this algorithm in order to visualize highly clustered data and has compared the results to that returned by the PCA. The author reported that by using the Generative Topographic Mapping algorithm, the three classes are better separated with a topographic map than with PCA.

Finally, although we did not apply segmentation following the localization of mitochondria, it is possible to link the work in Chapter 4 with that in Chapter 5 together. The experimental evidence indicates that both strategies should be used together in order to maximize the overall segmentation performance.

Bibliography

[1] D. Achlioptas and F. McSherry. Fast computation of low rank matrix approximations. *Journal of the ACM*, 54(2), April 2007.

[2] B. Afsari and P. S. Krishnaprasad. A family of ICA algorithms based on JD. *5th International Conference on ICA and BSS*, pages 441–445, 2004.

[3] Y. Al-Kofahi, W. Lassoued, W. Lee, and B. Roysam. Improved automatic detection and segmentation of nanoparticle nuclei in histopathology images. *IEEE Transactions on Biomedical Engineering*, 57(4):841–852, April 2010.

[4] R. Alomari, R. Allen, B. Sabata, and V. Chaudhary. Localization of tissues in high-resolution digital anatomic pathology images. *Proc. SPIE Medical imaging: Computer-Aided Diagnosis, FL,USA*, 7260, 2009.

[5] C. Alpert, A. Kahng, and S. Yao. Spectral partitioning: The more eigenvectors, the better. *Discrete Applied Mathematics*, 90:3–26, 1999.

[6] T. D. Alter and R. Basri. Extracting salient curves from images: an analysis of the saliency network. *International Journal of Computer Vision*, 27(1):51–69, 1998.

[7] C. Alzate and J. A. K. Suykens. Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA. *IEEE TPAMI*, 32(2):335–347, 2010.

[8] S. Amari, A. Cichocki, and H. H. Yang. Recurrent neural networks for blind separation of sources. *Int. Symposium on Nonlinear Theory and its Applications*, pages 37–42, 1995.

[9] M. R. Anderberg. *Cluster Analysis for Applications*. Academic Press Inc, 1973.

[10] J. August, K. Siddiqi, and S. W. Zucker. Contour fragment grouping and shared, simple occluders. *Computer Vision and Image Understanding*, 76(2):146–162, 1999.

[11] F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

[12] M. Barcena and A. J. Koster. Electron tomography in life science. *Seminars in Cell and Developmental Biology*, 20:920–930, 2009.

[13] E. M. L. Beale. Euclidean cluster analysis. *Bulletin of the International Statistical Institute: Proceedings of the 37th Session (London)*, pages 92–94, 1969.

[14] C. Becker, K. Ali, G. Knott, and P. Fua. Learning context cues for synapse segmentation. *IEEE Transactions on Medical Imaging*, 31(2):474–486, 2012.

[15] M.-A. Belabbas and P. Wolfe. Spectral methods in machine learning and new strategies for very large datasets. *Proceedings of National Academy of Sciences of the USA*, 106(2):369–374, 2009.

[16] A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1004–1034, 1995.

[17] A. J. Bell and T. J. Sejnowski. The independent components of natural scenes and edge filters. *Vision Research*, 37(23):3327–3338, 1997.

[18] J. Bell. *Machine Learning: Hands-On for Developers and Technical Professionals*. John Wiley and Sons, 2015.

[19] J. Bezdek, L. Hall, and L. Clarke. Review of MR image segmentation techniques using pattern recognition. *Medical Physics*, 20(4):1033–1048, 1993.

[20] T. D. Bie and N. Cristianini. Kernels methods for exploratory pattern analysis: a demonstration on text data. *Proceedings on IAPR and SSPR workshops*, Lisbon, August 2004.

[21] C. M. Bishop, M. Svensen, and C. K. I. Williams. GTM: A principled alternative to self-organizing map. *Proceedings of International Conference in Artificial neural Networks (ICANN'96)*, pages 165–170, 1996.

[22] C. M. Bishop, M. Svensen, and C. K. I. Williams. GTM: The generative topographic mapping. *Neural computation*, 10:215–234, 1998.

[23] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. *COLT*, 5:144–152, 1992.

[24] K. L. Boyer and S. Sarkar. *Perceptual Organization for Artificial Vision Systems*. Kluwer, 2000.

[25] J. Boykov and G. Funka-Lea. Graph cuts and efficient N-D image segmentation. *International Journal of Computer Vision*, 70(2):109–131, 2006.

[26] K. L. Briggman and D. D. Bock. Volume electron microscopy for neuronal circuit reconstruction. *Curr. Opin. Neurobiol.*, 22:154–161, 2012.

[27] J. L. Brooks. Traditional and new principles of perceptual grouping. *Oxford Handbook of Perceptual Organization*, Oxford University Press, 2015.

[28] F. Caillez and P. Kuntz. A contribution to the study of the metric and Euclidean structures of dissimilarities. *Psychometrika*, 61(2):241–253, 1996.

[29] G. A. Carpenter and S. Grossberg. *Adaptive Resonance Theory, The Handbook of Brain Theory and Neural Networks*. MIT Press, 2nd Edition, 2003.

[30] K. Cetina, P. Marquez-Neila, and L. Baumela. A comparative study of feature descriptors for mitochondria and synapse segmentation. *Proceedings of ICPR*, 2014.

[31] V. Cevher, M. F. Duarte, and R. G. Baraniuk. Sparse signal recovery using Markov random fields. *Advances in NIPS*, pages 257–264, 2009.

[32] D. Chang, L. Dooley, and J. E. Tuovinen. Gestalt theory in visual screen design - a new look at an old subject. *ACM International Conference Proceeding Series. Proceedings of the 7th World Conference on Computers in Education: Australian topics - Volume 8*, 26:5–12, 2002.

[33] H. Chang and D.-Y. Yeung. Robust path-based spectral clustering. *Pattern Recognition*, 41(1):191–203, 2008.

[34] N. Chater and G. Brown. Scale-invariance as a unifying psychological principle. *Cognition*, 69(3):B17–B24, 1999.

[35] W.-Y. Chen, Y. Song, H. Bai, C.-J. Lin, and E. Y. Chang. Parallel spectral clustering in distributed systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):568–586, March 2011.

[36] F. Chung. *Spectral graph theory. Volume 92 of the CMBS Regional Conference Series in Mathematics*. Conference Board of the Mathematical Sciences, Washington, 1997.

[37] D. Ciresan, A. Giusti, and L. M. Gambardalla. Deep neural networks segment neuronal membranes in electron microscopy images. *Advances in NIPS*, pages 2843–2851, 2012.

[38] A. G. Constantine and J. C. Gower. Graphical representation of asymmetric matrices. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 27(3):297–304, 1978.

[39] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[40] T. Cour, S. Yu, and J. Shi. *Ncut* demo software. *http://www.cis.upenn.edu/~jshi/software*.

[41] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling*. Chapman and Hall/CRC, 2nd edition, 2000.

[42] W. R. Crum. Spectral clustering and label fusion for 3d tissue classification: Sensitivity and consistency analysis. *Int. Conf. on Medical Image Understanding and Analysis*, 2008.

[43] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *Proceedings of CVPR*, 2005.

[44] M. Daszykowski, K. Kaczmarek, Y. Vander Heyden, and B. Walczak. Robust statistics in data analysis - a review: basic concepts. *Chemometrics and Intelligent Laboratory Systems*, 85(2):203–219, 2007.

[45] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications, Chapter 6: Point Location*. Springer-Verlag, 3rd Edition 2008.

[46] L. Deng and Y. Dong. *Deep Learning: Methods and Applications*. now Publishers Inc., 2013.

[47] W. Denk, K. L. Briggman, and M. Helmstaedter. Structural neurobiology: missing link to a mechanistic understanding of neural computation. *Nat. Rev. Neurosci.*, 13:351–358, 2012.

[48] A. Deshpande, L. Rademacher, S. Vempala, and G. Wang. Matrix approximation and projective clustering via volume sampling. *Symposium on Discrete Algorithms*, 2006.

[49] I. S. Dhillon, Y. Guan, and B. Kulis. Weighted graph cuts without eigenvectors: A multilevel approach. *IEEE TPAMI*, 29(11):1944–1957, 2007.

[50] J. Dietlmeier, O. Ghita, H. Duessmann, J. H. Prehn, and P. F. Whelan. Unsupervised mitochondria segmentation using recursive spectral clustering and adaptive similarity models. *Journal of Structural Biology*, 184(3):401–408, Elsevier, 2013.

[51] J. Dietlmeier, O. Ghita, and P. F. Whelan. On the projection similarity in line grouping. *Pattern Recognition Letters*, 51:50–56, Elsevier, 2015.

[52] J. Dietlmeier, O. Ghita, and P. F. Whelan. A new anticorrelation-based spectral clustering formulation. *Lecture Notes in Computer Science, Proceedings of Advanced Concepts for Intelligent Visioin Systems Conference*, 6915:139–149, Springer, 2011.

[53] J. Dietlmeier and P. F. Whelan. On the segmentation of intracellular subspaces: challenges and methods. *National Biophotonics and Imaging Platform Ireland ITC Workshop*, 2009.

[54] M. Donoser, M. Urschler, M. Hirzer, and H. Bischof. Saliency driven total variation segmentation. *ICCV*, 2009.

[55] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering large graphs via the singular value decomposition. *Machine Learning*, 56:9–33, 2004.

[56] P. Drineas and R. Kannan. Pass efficient algorithms for approximating large matrices. *Proceedings of the 14th Annual Symposium on Discrete Algorithms, Baltimore, MD*, pages 223–232, 2003.

[57] P. Drineas and M. W. Mahoney. On the Nyström method for approximating a Gram matrix for improved kernel-based learning. *JMLR*, 2005.

[58] R. O. Duda and P. E. Hart. Pattern classification and scene analysis. John Wiley and Sons., New York, 1973.

[59] S. Dudoit and J. Fridlyand. A prediction-based resampling method for estimating the number of clusters in a data set. *Genome Biology*, 3(7), 2002.

[60] S. Duffner. Face image analysis with convolutional neural networks. *PhD Thesis*, GRIN verlag, 2008.

[61] R. Durbin and D. Willshaw. An analogue approach to the travelling salesman problem using an elastic net method. *Nature*, 326:689–691, 1987.

[62] W. H. Ehrenstein, L. Spillmann, and V. Sarris. Gestalt issues in modern neuroscience. *Axiomathes*, 13(3):433–458, 2003.

[63] P. Estevez, R. Flores, and C. Perez. Color image segmentation using fuzzy min-max neural networks. *Proc. IEEE Int. Joint Conf. Neural Networks (IJCNN'05), Montreal, Quebec, Canada*, 5:3052–3057, 2005.

[64] B. S. Everitt, S. Landau, M. Leese, and D. Stahl. *Cluster Analysis, 5th Edition*. John Wiley and Sons, Ltd, 2011.

[65] D. W. Fawcett. *The Cell*. http://www.ascb.org, 1981.

[66] J. J. Fernandez. Computational methods for electron tomography. *Micron*, 43:1010–1030, 2012.

[67] W. Fiers, R. Beyaert, W. Declercq, and P. Vandenabeele. More than one way to die: apoptosis, necrosis and reactive oxygene damage. *Oncogene*, 18(54):7719–7730, 1999.

[68] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the Nyström method. *IEEE TPAMI*, 26(2):214–225, 2004.

[69] C. Fowlkes, D. Martin, and J. Malik. Learning affinity functions for image segmentations: combining patch-based and gradient-based approaches. *CVPR*, 2003.

[70] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.

[71] A. Frieze, R. Kannan, and S. Vempala. Fast monte-carlo algorithms for finding low-rank approximations. *Journal of ACM*, 51(6):1025–1041, 2014.

[72] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Journal Biological Cybernetics*, 36(4):193–202, 1980.

[73] K. Fukushima. Analysis of the process of visual pattern recognition by the neocognitron. *Neural Networks*, 2:413–421, 1989.

[74] C. Garcia and M. Delakis. Convolutional face finder. a neural architecture for fast and robust face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1408–1423, 2004.

[75] J. M. Geusebroek and A. W. M. Smeulders. A minimum cost approach for segmenting networks of lines. *International Journal of Computer Vision*, 43(2):99–111, 2001.

[76] O. Ghita, J. Dietlmeier, and P. F. Whelan. Automatic segmentation of mitochondria in EM data using pairwise affinity factorization and graph-based contour searching. *IEEE Transactions on Image Processing*, 23(10):4576–4586, 2014.

[77] A. Ghodsi. Dimensionality reduction, a short tutorial. *Technical report*, Department of Statistics and Actuarial Science, University of Waterloo, Canada, 2006.

[78] F. D. Gibbons and F. P. Roth. Judging the quality of gene expression-based clustering methods using gene annotation. *Genome Research*, 12(10):1574–1581, 2002.

[79] R. Giuly, M. E. Martone, and M. H. Ellisman. Method: Automatic segmentation of mitochondria utilizing patch classification, contour pair classification, and automatically seeded level sets. *BMC Bioinformatics*, 13:29 doi:10.1186/1471-2105-13-29, 2012.

[80] G. H. Golub and C. F. Van Loan. *Matrix Computations, 3rd edition*. Johns Hopkins University Press, Baltimore, MD, 1996.

[81] G. Gorrell. Generalized Hebbian algorithm for incremental singular value decomposition in natural language processing. *EACL*, 2006.

[82] S. Grossberg. *Recurrent Neural Networks*. Scholarpedia, 8(2), 2013.

[83] S. Grossberg and E. Mingolla. Neural dynamics of form perception: boundary completion, and illusory figures, and neon color spreading. *Psychological Review*, 92:173–211, 1985.

[84] S. Guha, R. Rastogi, and K. Shim. An efficient clustering algorithm for large databases. *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 73–84, 1998.

[85] Y. Guo, B. Gong, S. Levesque, T. Manfredi, and Y. Sun. Automated detection and delineation of mitochondria in electron micrographs of human skeletal muscles. *Microscopy Research and Technique*, 63(3):133–139, 2004.

[86] G. Guy and G. Medioni. Inferring global perceptual contours from local features. *International Journal of Computer Vision*, 20(1):113–133, 1996.

[87] K. G. Hales. Mitochondrial fusion and division. *Nature Education*, 3(9):12, 2003.

[88] L. O. Hall, A. M. Bensaid, L. P. Clarke, and R. P. Velthuizen. A comparison of neural network and fuzzy clustering techniques in segmenting magnetic resonance images of the brain. *IEEE Transactions on Neural Networks*, 3(5):672–682, 1992.

[89] R. J. Hathaway and J. C. Bezdek. NERF C-Means: Non-Euclidean Relational Fuzzy Clustering. *Pattern Recognition*, 27(3):429–437, 1994.

[90] F. Heitger, R. von der Heydt, E. Peterhans, L. Rosenthaler, and O. Kuebler. Simulation of neural contour mechanisms: representing anomalous contours. *Image and Vision Computing*, 16:407–421, 1998.

[91] M. Helmstaedter. Cellular-resolution connectomics: challenges of dense neural circuit reconstruction. *Nature Methods*, 10:501–507, 2013.

[92] H. Hiary, R. S. Alomari, and V. Chaudhary. Segmentation and localization of whole slide images using unsupervised learning. *IET image processing*, 7(5):464–471, 2013.

[93] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, May 2006.

[94] S. Hirano and S. Tsumoto. Hierarchical clustering of asymmetric proximity data based on the indiscernibility-level. *Proceedings on IEEE International Conference on Granular Computing 2008*, pages 275–280, August 2008.

[95] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in cat's visual cortex. *Journal of Physiology*, 160(1):106–154, 1962.

[96] A. Hyvarinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4):411–430, 2000.

[97] V. Jain, H. S. Seung, and S. C. Turaga. Machines that learn to segment images: a crucial technology for connectomics. *Curr. Opin. Neurobiol.*, 20(5):653–666, 2010.

[98] J.-H. Jang and K.-S. Hong. Fast line segment grouping method for finding globally more favorable line segments. *Pattern Recognition*, 35(10):2235–2247, 2002.

[99] X. Jiang. Asymmetric principal component and discriminant analyses for pattern classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(5):931–937, 2009.

[100] A. Johri and M. F. Beal. Mitochondrial dysfunction in neurodegenerative diseases. *Journal of Pharmacology and Experimental Therapeutics*, 342(3):619–630, 2012.

[101] I. T. Jolliffe. *Principal Component Analysis, 2nd ed.* Springer-Verlag, 2002.

[102] N. E. Karoui and A. d'Aspremont. Second order accurate distributed eigenvector computation for extremely large matrices. *Electronic Journal of Statistics*, 4:1345–1385, 2010.

[103] V. Kaynig, T. Fuchs, and J. M. Buhmann. Neuron geometry extraction by perceptual grouping in ssTEM images. *CVPR*, pages 2902–2909, 2010.

[104] A. R. Kelly and E. R. Hancock. Grouping line segments using eigenclustering. *British Machine Vision Conference (BMVC)*, 2000.

[105] T. H. Kim, K. M. Lee, and S. U. Lee. Learning full pairwise affinities for spectral segmentation. *IEEE TPAMI*, 35(7):1690–1703, 2013.

[106] G. Knott and C. Genoud. Is EM dead? *Journal of Cell Science*, 126:4545–4552, 2013.

[107] A. V. Knyazev and M. E. Argentati. Principal angles between subspaces in a A-based scalar product: algorithms and perturbation estimates. *SIAM J. Sci. Comput.*, 23(6):2009–2041, 2002.

[108] K. Koffa. *Principles of Gestalt psychology*. London, Routledge and Kegan Paul, 1935.

[109] O. Z. Kraus, J. L. Ba, and B. J. Frey. Classifying and segmenting microscopy images with deep multiple instance learning. *Bioinformatics*, 32(12):52–59, 2016.

[110] J. R. Kremer, D. N. Mastronarde, and J. R. McIntosh. Computer visualization of three-dimensional image data using IMOD. *Journal of Structural Biology*, 116(1):71–76, 1996.

[111] A. Kreshuk, C. N. Straehle, C. Sommer, U. Koethe, M. Cantoni, G. Knott, and F. A. Hamprecht. Automated detection of synaptic contacts in nearly isotropic serial electron microscope images. *PLos ONE*, 6:e24899, 2011.

[112] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. *Proceedings of Neural Information Processing Systems (NIPS)*, 2012.

[113] R. Kumar, A. Vazquez-Reina, and H. Pfister. Radon-like features and their applications to connectomics. *Proc. Int. Conference on Computer Vision and Pattern Recognition Workshops*, pages 186–193, 2010.

[114] G. N. Lance and W. T. Williams. A General Theory of Classificatory Sorting Strategies. 1. Hierarchical Systems. *Computer Journal*, 9:373–380, 1967.

[115] T. Lange, M. Braun, V. Roth, and J. M. Buhmann. Stability-based model selection. *Proceedings of Conference on Neural Information Processing Systems*, 2003.

[116] P. Langfelder, B. Zhang, and S. Horvath. Defining clusters from a hierarchical cluster tree; the Dynamic Tree Cut package for R. *Bioinformatics*, 24, 1943.

[117] J. Laub and K. R. Miller. Feature discovery in non-metric pairwise data. *JMLR*, 5:801–818, 2004.

[118] J. Laub, V. Roth, J. Buhmann, and K. Muller. On the information and representation of non-Euclidean pairwise data. *Pattern Recognition*, 39:1815–1826, 2006.

[119] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back. Face recognition: A convolutional neural network approach. *IEEE Transactions on Neural Networks*, 8:98–113, 1997.

[120] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278–2324, 1998.

[121] S. M. Leena and V. K. Govindan. Convolutional neural network based segmentation. *ICIP 2011, CCIS 157, Springer-Verlag Berlin Heidelberg 2011*, pages 190–197, 2011.

[122] S. M. Leena and V. K. Govindan. Enhanced CNN based electron microscopy image segmentation. *Cybernatics and Information Technologies*, 12(2):84–97, 2012.

[123] M. Leone, Sumedha, and M. Weigt. Unsupervised and semi-supervised clustering by message passing: soft-constraint affinity propagation. *The European Physical Journal B*, 66:125–135, 2008.

[124] W. Li and C. D. Gilbert. Global contour saliency and local colinear interactions. *Journal of Neurophysiology*, 88:2846–2856, 2002.

[125] W. Li, Q. Rao, X. Chen, G. Li, and D. Zhang. Segmentation of mitochondria based on SEM images. *Proceedings of IEEE Int. Conf. on Mechatronics and Automation (ICMA)*, 2016.

[126] R. Linsker. Self-organization in a perceptual network. *Computer*, 21:105–117, 1988.

[127] R. Linsker. How to generate ordered maps by maximizing the mutual information between input and output signals. *Neural Computation*, 1:402–411, 1989.

[128] H. Liu, L. Jiao, and F. Zhao. Non-local spatial spectral clustering for image segmentation. *Neurocomputing*, 74:461–471, 2010.

[129] R. Liu and H. Zhang. Segmentation of 3D meshes through spectral clustering. *Proceedings of Pacific Graphics*, 2004.

[130] D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, 1987.

[131] D. G. Lowe. Perceptual organization and visual recognition. Kluwer Academic Publishers, Norwell, MA, USA, 1985.

[132] A. Lucchi, K. Smith, R. Achanta, and P. Knott, G Fua. Supervoxel-based segmentation of mitochondria in em image stacks with learned shape features. *IEEE Trans. Med. Imaging*, 31:474–486, 2012.

[133] A. Lucchi, K. Smith, R. Achanta, V. Lepetit, and P. Fua. A fully automated approach to segmentation of irregularly shaped cellular structures in EM images. *MICCAI*.

[134] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[135] S. Mahamud, L. R. Williams, K. K. Thornber, and K. Xu. Segmentation of multiple salient closed contours from real images. *IEEE TPAMI*, 25(4):433–444, 2003.

[136] J. Malik, S. Belongie, T. Leung, and T. Shi. Contour and texture analysis for image segmentation. *Perceptual Organization for Artificial Vision Systems*, Kluwer, 2000.

[137] J. Malik, S. Belongie, J. Shi, and T. Leung. Textons, contours and regions: Cue integration in image segmentation. *IEEE International Conference on Computer Vision*, Corfu, Greece, September 1999.

[138] P. Marquez-Neila, L. Baumela, J. Gonzalez-Soriano, J. R. Rodriguez, J. DeFelipe, and A. ferchan Perez. A fast method for the segmentation of synaptic junctions and mitochondria in serial electron microscopy images of the brain. *Neuroinformatics*, 14(2):235–250, 2016.

[139] B. J. Marsh, D. N. Mastronarde, K. F. Buttle, K. E. Howell, and J. R. McIntosh. Organellar relationships in the Golgi region of the pancreatic beta cell line, HIT-T15, visualized by high resolution electron tomography. *PNAS USA*, 98(5):2399–2406, 2001.

[140] S. Marsland. *Machine Learning: An Algorithmic Perspective. Second Edition*. Chapman and Hall, 2015.

[141] A. Martinez-Sanchez, I. Garcia, and J. J. Fernandez. A differential structure approach to membrane segmentation in electron tomography. *Journal of Structural Biology*, 175:372–383, 2011.

[142] T. Martiriggiano, M. Leo, T. D'Orazio, and A. Distante. Face recognition by kernel independent component analysis. *Lecture Notes in Computer Science*, 3533:55–58, 2005.

[143] J. J. McCann. Image processing analysis of traditional gestalt vision experiments. *AIC Proceedings, Rochester*, pages 375–378, 2002.

[144] W. S. McCulloch and W. Pitts. A logical calculus of ideas imminent in nervous activity. *Bulletin of Mathematics Biophysics*, 5:115–133, 1943.

[145] M. L. Minsky and S. A. Papert. *Perceptrons*. MIT Press, Cambridge, MA, 1969.

[146] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012.

[147] F. C. Monteiro and A. C. Campilho. Spectral methods in image segmentation: a combined approach. *LNCS*, 3523:191–198, 2005.

[148] J. H. Morra, Z. Tu, L. G. Apostolova, A. E. Green, A. W. Toga, and P. M. Thomson. Comparison of AdaBoost and Support Vector Machines for detecting Alzheimer's disease through automated hippocampal segmentation. *IEEE Transactions on Medical Imaging*, 29(1):30–43, 2010.

[149] K.-R. Mueller, T. Adali, K. Fukumizu, J. C. Principe, and S. Theodoridis. Machine learning. *IEEE Signal Processing Magazine*, 14, July 2013.

[150] R. Multihac and M. M. Van Hulle. A comparative survey on adaptive neural network algorithms for independent component analysis. *Rom. Reports in Phys*, 55(1):49–74, 2003.

[151] E. U. Mumcuoglu, R. Hassanpour, S. F. Tasel, G. Perkins, M. E. Martone, and M. N. Gurcan. Computerized detection and segmentation of mitochondria in electron microscope images. *Journal of Microscopy*, 246(3):248–265, 2012.

[152] P. F. M. Nacken. A metric for line segments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12):1312–1318, 1993.

[153] R. Narasimha, H. Ouyang, A. Gray, S. W. McLaughlin, and S. Subramaniam. Automatic joint classification and segmentation of whole cell 3D images. *Pattern Recognition*, 42(6):1067–1079, 2009.

[154] S. Navlakha, J. Suhan, A. L. Barth, and Z. Bar-Joseph. A high-throughput framework to detect synapses in electron microscopy images. *Bioinformatics*, 29(13):i9–i17, 2013.

[155] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *NIPS*, 14:849–856, 2001.

[156] H. Nguyen and Q. Ji. Shape-driven three-dimensional watersnake segmentation of biological membranes in electron tomography. *IEEE Transactions on Medical Imaging*, 27(5):616–628, 2008.

[157] F. Ning, D. Delhomme, and Y. LeCun. Toward automatic phenotyping of developing embryos from videos. *IEEE Transactions on Image Processing*, 14(9):1360–1371, 2005.

[158] A. B. Noske, A. J. Costin, G. P. Morgan, and B. J. Marsh. Expedited approaches to whole cell electron tomography and organelle mark-up in situ in high-pressure frozen pancreatic islets. *Journal of Structural Biology*, 161(3):298–313, 2008.

[159] J. Novembre and M. Stephens. Interpreting principal component analyses of spatial population genetic variation. *Nature Genetics*, 40:646–649, 2008.

[160] S. J. Nowlan and J. C. Platt. Convolutional neural network hand tracker. *Advances in Neural Information Processing Systems*, 7, 1995.

[161] N. Paragios. Computer vision research: The deep depression. Available online through *www.linkedin.com/pulse/*, June 2016.

[162] C. Park, J. Z. Huang, J. X. Ji, and Y. Ding. Segmentation, inference and classification of partially overlapping nanoparticles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):669–681, March 2013.

[163] C. J. Peddie and L. M. Collinson. Exploring the third dimension: Volume electron microscopy comes of age. *Micron*, 61:9–19, 2014.

[164] E. Pekalska, A. Harol, R. P. W. Duin, B. Spillmann, and H. Bunke. Non-Euclidean or non-metric measures can be informative. *LNCS*, 4109:871–880, 2006.

[165] G. A. Perkins, M. G. Sun, and T. G. Frey. Correlated light and electron microscopy/electron tomography of mitochondria in situ. *Methods in Enzymology*, 456:29–52, 2009.

[166] P. Perona and W. T. Freeman. Factorization approach to grouping. *ECCV*, pages 655–670, 1998.

[167] J. Podani and I. Miklós. Resemblance coefficients and the horseshoe effect in principal component analysis. *Ecology*, 83(12):3331–3343, 2002.

[168] D. M. W. Powers. Evaluation: from Precision, Recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technology*, 2(1):37–63, 2011.

[169] M. Rehm, H. J. Huber, C. T. Hellwig, S. Anguissola, H. Dussmann, and J. H. M. Prehn. Dynamics of outer mitochondrial membrane permeabilization during apoptosis. *Cell Death and Differentiation*, 16(4):613–623, 2009.

[170] J. L. Rodgers and W. A. Nicewander. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1):59–66, 1988.

[171] R. Rojas. Kohonen networks. *Neural Networks*, pages 391–412, Springer, 1996.

[172] O. Ronnenberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *MICCAI, Springer, LNCS*, 9351:234–241, 2015.

[173] F. Rosenblatt. The perceptron: a probabilistic model for information storage and retrieval in the brain. *Psychological Review*, 65:386–408, 1958.

[174] F. Rosenblatt. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan, Washington, DC, 1962.

[175] V. Roth, J. Laub, J. M. Buhmann, and K.-R. Mueller. Going metric: denoising pairwise data. *Advances in NIPS*, 15:817–824, 2002.

[176] V. Roth, J. Laub, M. Kawanabe, and J. M. Buhmann. Optimal cluster preserving embedding of nonmetric proximity data. *IEEE TPAMI*, 25(12):1540–1551, 2003.

[177] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. *Parallel Distributed Processing: Explorations in the Microstructures of Cognition, MIT Press 1986*, pages 318–362, 1986.

[178] Z. Saidane. *Image and video text recognition using convolutional neural networks*. Lambert Academic Publishing, 2011.

[179] G. Sanguinetti. Dimensionality reduction in clustered data sets. *IEEE TPAMI*, 30(3):535–540, 2008.

[180] J. Schmidhuber. Deep learning in neural networks: an overview. *Neural Networks*, 61:85–117, 2015.

[181] B. Schoelkopf, A. Smola, and K.-R. Mueller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[182] B. Schoelkopf and A. J. Smola. *Learning with Kernels*. MIT press, Cambridge, MA, 2002.

[183] M. Scholz, M. Fraunholz, and J. Selbig. Nonlinear principal component analysis: Neural networks models and applications. *Principal Manifolds for Data Visualization and Dimension Reduction*, pages 44–67, Springer, 2008.

[184] M. Seyedhosseini, M. Ellisman, and T. Tasdizen. Segmentation of mitochondria in electron microscopy images using algebraic curves. *Proc. of the 10th IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 860–863, 2013.

[185] A. Sha'ashua and S. Ullman. Structural saliency: the detection of globally salient structures using a locally connected network. *ICCV*, pages 321–327, December 1988.

[186] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning. From Theory to Algorithms*. Cambridge University Press, 2014.

[187] L. Shamir, J. D. Delaney, N. Orlov, D. M. Eckley, and I. G. Goldberg. Pattern recognition software and techniques for biological image analysis. *PLoS Computational Biology*, 6(11):1–10, 2010.

[188] R. Shenoy, M.-C. Shih, and K. Rose. Segmentation of cells in electron microscopy images through multimodal label transfer. *Proceedings of IEEE ICIP*, 2015.

[189] J. Shi and J. Malik. Normalized cuts and image segmentation. *CVPR*, 1997.

[190] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE TPAMI*, 22(8):888–905, 2000.

[191] Y. Shi. A structural view of mitochondria-mediated apoptosis. *Nature Structural Biology*, 8(5):394–401, May 2001.

[192] V. Silva and B. Tenenbaum. Unsupervised learning of curved manifolds. *Nonlinear Estimation and Classification. Lecture Notes in Statistics. Springer Verlag*, 171:453–465, 2003.

[193] K. Smith, A. Carleton, and V. Lepetit. Fast ray features for learning irregular shapes. *Proceedings of ICCV*, 2009.

[194] N. Srebro and T. Jaakkola. Linear dependent dimensionality reduction. *Advances in NIPS*, 2003.

[195] J. S. Stahl and S. Wang. Globally optimal grouping for symmetric closed boundaries by combining boundary and region information. *IEEE TPAMI*, 30(3):395–411, 2008.

[196] K. Stepka. Automatic detection and segmentation of exosomes in transmission electron microscopy. *ECCV 2016 Workshops, LNCS*, 9913:318–325, 2016.

[197] G. W. Stewart. *Matrix Algorithms, Volume II: Eigensystems*. SIAM, 2001.

[198] M. F. Stollenga, W. Byeon, and M. Liwicki. Parallel multi-dimensional LSTM, with application to fast biomedical volumetric image segmentation. *Proceeding of the 28th Int. Conf. on Neural Information Processing Systems*, pages 2998–3006, 2015.

[199] M. G. Sun, J. Williams, C. Munoz-Pinedo, G. A. Perkins, J. M. Brown, M. H. Ellisman, D.-R. Green, and T. G. Frey. Correlated three-dimensional light and electron microscopy reveals transformation of mitochondria during apoptosis. *Nature Cell Biology*, 9(9):1057–1065, 2007.

[200] C. T. N. Suzuki and A. X. Falcão. Livewire: 2d interactive tool for medical image segmentation. Available online at http://www.ic.unicamp.br/∼afalcao/livewire/.

[201] A. Talwalkar, S. Kumar, and H. Rowley. Large-scale manifold learning. *CVPR*, 2008.

[202] T. Tasdizen, E. Jurrus, and R. T. Whitaker. Non-uniform illumination correction in transmission electron microscopy images. *MICCAI Workshop on Microscopic Image Analysis with Applications in Biology (MIAAB08)*, 2008.

[203] F. B. Tek, T. Kroeger, S. Mikula, and F. A. Hamprecht. Automated cell nucleus detection for large-volume electron microscopy of neural tissue. *IEEE International Symposium on Biomedical Imaging*, 2014.

[204] S. Thiberge, A. Nechustan, D. Sprinzak, O. Gileadi, V. Behar, O. Zik, Y. Chowers, S. Michaeli, J. Schlessinger, and E. Moses. Scanning electron microscopy of cells and tissues under fully hydrated conditions. *PNAS*, 101(10):3346–3351, 2004.

[205] V. Thong Ta, O. Lezoray, A. Elmoataz, and S. Schüpp. Graph-based tools for microscopic cellular image segmentation. *Pattern Recognition*, 42:1113–1125, 2009.

[206] X. Tian, X. Zhang, X. Deng, and S. Chen. Multiway kernel independent component analysis based on feature samples for batch process monitoring. *Neurocomputing*, 72:1584–1596, 2009.

[207] H. T. Tran, D. A. Romanov, and R. J. Levis. Control goal selection through anticorrelation analysis in the detection space. *J. Phys. Chem. A*, 110:10558–10563, 2006.

[208] F. Tung, A. Wong, and D. A. Clausi. Enabling scalable spectral clustering for image segmentation. *Pattern Recognition*, 43:4069–4076, 2010.

[209] G. Tzortzis and A. Likas. The global kernel k-means clustering algorithm. *Neural Networks. IEEE IJCNN*, pages 1977–1984, 2008.

[210] W. R. Ullman. Three dimensional object recognition. *Cold Spring Harbor Symposia on Quantitative Biology*, pages 889–898, 1990.

[211] D. Unay and B. Gosselin. Artificial neural network-based segmentation and apple grading by machine vision. *IEEE ICIP, Genoa, Italy*, 2:630–633, 2005.

[212] H. Valpola. From neural PCA to deep unsupervised learning. *Advances in Independent Component Analysis and Learning Machines*, pages 143–171, Elsevier, 2015.

[213] M. M. Van Hulle. Self-organizing maps. *Handbook of Natural Computing*, pages 585–622, Springer, 2012.

[214] F. Van Veen. The neural network zoo. *The Asimov Institute*, Available online at http://www.asimovinstitute.org/neural-networks-zoo. September 2016.

[215] F. H. T. Vieira and L. L. Lee. A neural architecture based on the adaptive resonance theory and recurrent neural networks. *International Journal of Computer Science and Applications*, 4(3):45–56, 2007.

[216] S. Vitaladevuni, Y. Mishchenko, A. Genkin, D. Chklovskii, and K. Harris. Mitochondria detection in electron microscopy images. *Workshop on Microscopic Image Analysis Applications Biology (MIAAB)*, New York, USA, 2008.

[217] J. Wagemans, M. Kubovy, M. A. Peterson, J. H. Elder, S. E. Palmer, and M. Singh. A century of Gestalt psychology in visual perception: I. perceptual grouping and figure-ground organization. *Psychological Bulletin*, 138(6):1172–1217, 2012.

[218] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Speech and Signal Processing*, 37:328–339, 1989.

[219] Z. Wang. A new approach for segmentation and quantification of cells or nanoparticles. *IEEE Transactions on Industrial Informatics*, 12(3):962–971, June 2016.

[220] Y. Weiss. Segmentation using eigenvectors: a unifying view. *ICCV*, 1999.

[221] M. Wertheimer. Untersuchungen zur Lehre von der Gestalt II. *Psychologische Forschung*, 4:301–350, 1923.

[222] G. Westheimer. Gestalt theory reconfigured: Max Wertheimer's anticipation of recent developments in visual neurosciences. *Perception*, 28(1):5–15, 1999.

[223] C. K. I. Williams. On a connection between kernel PCA and metric multidimensional scaling. *Machine Learning*, 46(1-3):11–19, 2002.

[224] C. K. I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. *NIPS*, pages 682–688, 2000.

[225] L. R. Williams and D. W. Jacobs. Stochastic completion fields: A neural model of illusory contour shape and salience. *Neural Computation*, 9(4):837–858, 1997.

[226] M. Yamasaki, Y.-W. Chen, and G. Xu. Separating reflections from images using kernel independent component analysis. *ICPR*, 3:194–197, 2006.

[227] H. Yang and N. Ahuja. Automatic segmentation of granular objects in images: Combining local density clustering and gradient-barrier watershed. *Pattern Recognition*, 2014.

[228] J. Yang, X. Gao, D. Zhang, and J. Yang. Kernel ICA: an alternative formulation and its application to face recognition. *Pattern Recognition*, 38(10):1784–1787, 2005.

[229] C. Zhang, J. Yarkony, and F. A. Hamprecht. Cell detection and segmentation using correlation clustering. *LNCS*, 2014.

[230] N. Zheng and J. Xue. *Statistical Learning and Pattern Analysis for Image and Video Processing*. Springer, 2009.

[231] K. H. Zou, S. K. Warfield, A. Bharatha, C. M. Tempany, M. R. Kaus, S. J. Haker, W. Wells, F. A. Jolesz, and R. Kikinis. Statistical validation of image segmentation quality based on a spatial overlap index. *Academic Radiology Journal*, 11(2):178–189, Elsevier, 2004.